

Ergänzungen zur Klangregelung

Version 1.3

Jürgen Döffinger

12. Februar 2014

Inhaltsverzeichnis

Inhaltsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VI
Formelzeichenverzeichnis	VII
1. Einleitung	1
2. Übersicht über die Änderungen	1
3. Grundlagen	2
3.1. Der Clipper	2
3.2. Der Echo-Effekt	5
3.3. Der Dynamic Processor	6
3.3.1. Die Detektoren zur Ermittlung des Effektiv- und Spitzenwertes	7
3.3.2. Die Kennlinie des Dynamic Processors	9
3.3.3. Die Ansprech- und Rücklaufzeit	26
3.3.4. Abtastratenreduktion	27
4. Die Software	28
4.1. Das Audio Signal Processing Program (ASPP)	28
4.2. Das Control and User Interface Program (CUIP)	33
4.3. Das Tone Control Graphical User Interface Program (TCGUIP)	39
4.4. Das Tone Control Webinterface Program (TCWIP)	41
5. Die Messergebnisse	42
6. Tests und erkannte Fehler	42
7. Zusammenfassung und Ausblick	42
Literaturverzeichnis	43
Anhang	

Abbildungsverzeichnis

1. Kennlinie des Clippers	3
2. Signal ohne Begrenzung	3
3. Signal mit Begrenzung	3
4. Signal schwach begrenzt	4
5. Signal stark begrenzt	4
6. Signal wie in Bild 4 ohne Clipper	5
7. Signalflussgraph des Echo-Effektes	5
8. Signalflussgraph Dynamic Processor	7
9. Kennlinie von Eingangspegel zu Ausgangspegel ohne Dynamic Processor	11
10. Kennlinie der einzelnen Elemente des Dynamic Processors	12
11. Kennlinie des Dynamic Processors	13
12. Kennlinie des Dynamic Processors mit angepassten Gleichungen	14
13. Wirkung von Kompressor und Expander	15
14. B-B-Kurve	19
15. Soft-Knee - Compressor/Expander	23
16. Dynamic Processor Kennlinie (vollständig)	24
17. Verschachtelungstechnik	27
18. TCGUIP - Dynamic processor window	39
19. TCGUIP - Main window	40
20. TCWIP - Benutzeroberfläche	41

Tabellenverzeichnis

1. Übersicht über die Änderungen der Klangregelung	1
2. Zuordnung der Funktion zu den Kompressionsfaktoren (Ratio)	9
3. Zuordnung der Symbole zu den Parametern der Funktionen des Dynamic Processors	10

Abkürzungsverzeichnis

AOG	automatischer Ausgangsverstärkungsfaktor (Auto Output Gain)
ASPP	Audiosignalverarbeitungsprogramm (Audio Signal Processing Program)
AU	Audio Unit
CU	Control Unit
CUIP	Control and User Interface Program
I ² C	Inter-Integrated-Circuit
LCD	Flüssigkristallanzeige (Liquid Crystal Display)
LUT	Nachschlagetabelle (LookUp Table)
MOG	manueller Ausgangsverstärkungsfaktor (Manuell Output Gain)
NaN	keine Zahl (Not a number)
TCGUIP	Tone Control Graphical User Interface Program
TCWIP	Tone Control Webinterface Program

Formelzeichenverzeichnis

Formelzeichen

AT	Koeffizient der Ansprechzeit (Attack time)
B	BERNSTEINSche Grundpolynome
C	Kompressor (Compressor)
D	Verzögerungskoeffizient
E	Expander (Expander)
f_s	Abtastfrequenz (Sample frequency)
$f(n)$	Steuerfaktor des Dynamic Processors
FS	Vollaussteuerung (Full scale)
g	linearer Verstärkungsfaktor
G	logarithmischer Verstärkungsfaktor in dB
G_a	logarithmischer Verstärkungsfaktor des Auto Output Gain in dB
G_{en}	logischer Wert zum Aktivierungszustand des AOG
G_D	logarithmischer Verstärkungsfaktor des Dynamic Processors in dB
G_K	logarithmischer Verstärkungsfaktor der Kennlinie des Dynamic Processors in dB
G_m	logarithmischer Verstärkungsfaktor des Manuell Output Gain in dB
L	Begrenzer (Limiter)
N	Rauschunterdrückung (Noise-Gate)
N_w	Wortbreite des diskretisierten Audiosignals
P	Punkt (Point)
\vec{r}	Raumkurve der BERNSTEIN-BÉZIER-Darstellung
R	Kompressionsfaktor (Ratio)
RT	Koeffizient der Rücklaufzeit (Release time)
S	Steigung (Slope)
T	Schwelle (Threshold)
t_a	Ansprechzeit (Attack Time) in Millisekunden
TAV	Koeffizient der Mittelungszeit
t_M	Mittelungszeit in Millisekunden
t_r	Rücklaufzeit (Release Time) in Millisekunden
$x(n)$	zeitdiskretes Eingangssignal
$y(n)$	zeitdiskretes Ausgangssignal

Indizes

<i>D</i>	Dynamic Processor
<i>E</i>	Echo Effekt
<i>en</i>	aktiviert/deaktiviert (enabled/disabled)
<i>K</i>	Breite des Soft-Knee in dB
<i>l</i>	linker Kanal (left channel)
<i>left</i>	linker Kanal (left channel)
<i>peak</i>	Spitzenwert (Peak)
<i>r</i>	rechter Kanal (right channel)
<i>R</i>	Kompressionsfaktor (Ratio)
<i>right</i>	rechter Kanal (right channel)
<i>RMS</i>	Effektivwert (Root Mean Square)
<i>S</i>	Steigung (Slope)
<i>T</i>	Schwelle (Threshold)

1. Einleitung

In der Abschlussarbeit *Implementierung einer Klangregelung durch Shelving-Filter auf einem Signalprozessor mit Steuerung der Parameter über einen Mikrocontroller* zur Erlangung des akademischen Grades Bachelor of Engineering sprach der Verfasser von der Möglichkeit auf einfache Weise die Klangregelung mit weiteren Filtern, Effekten oder ähnlichem zu erweitern. Dies soll hier gezeigt werden. Dazu wird in den folgenden Abschnitten die Grundlagen für einen Clipper, Echo-Effekt und einen Dynamic Processor zusammengetragen, um im Anschluss zu zeigen, wie diese in die bestehende Klangregelung eingearbeitet wurden.

2. Übersicht über die Änderungen

In der schriftlichen Ausarbeitung zur Bachelorarbeit schrieb der Verfasser der Arbeit immer von der Erweiterbarkeit der Klangregelung. Um diese nachzuweisen, wurde der ursprünglichen Version ein Echo-Effekt und ein Dynamic Processor hinzugefügt. Diese werden im Abschnitt 3 näher beschrieben. Die folgende Tabelle gibt eine Übersicht über die Änderungen bis zur aktuellen Version 1.3.

Version	Änderungen
1.1	- Clipper hinzugefügt
1.2	- Echo-Effekt hinzugefügt
1.3	- Limiter entfernt - Dynamic Processor hinzugefügt

Tabelle 1: Übersicht über die Änderungen der Klangregelung

3. Grundlagen

Im Abschnitt Grundlagen sollen die technischen und mathematischen Grundlagen der hinzugefügten Elemente aufgezeigt werden. Bei den folgenden Erläuterungen wird nicht der zeitliche Ablauf berücksichtigt und es wird auch nicht auf die einzelnen Änderungen in den Versionen eingegangen, sondern die Änderungen im Gesamten betrachtet.

3.1. Der Clipper

Der Limiter der für die Begrenzung zu großer Signale verantwortlich war, konnte kurze Impulse mit großer Amplitude nicht schnell genug abfangen. Dies liegt daran, dass die Ansprechzeit t_a nicht weiter verringert werden konnte, da diese von der Abtastfrequenz f_s abhängig ist. Abgesehen davon bringt das Verringern der Ansprechzeit auch Nachteile für das Klangbild.

Aus dem zuvor genannten Grund wurde nach dem Limiter ein Clipper hinzugefügt. Dieser fängt die Impulse ab in dem er auf Full scale (FS) begrenzt. Somit kommt es nicht zu einem Überlauf, welcher als Knackgeräusche hörbar ist, sondern zu Verzerrungen, welche bei den angesprochenen kurzen Impulsen kaum bis nicht wahrgenommen werden. Sollte jedoch der Eingangspegel größer werden, welcher dem Clipper zugeführt wird, werden auch hier die Verzerrungen hörbar.

Die Klangregelung arbeitet mit einer Wortbreite $N_w = 16$ bit. Somit ergibt sich für Full scale $FS = 2^{15} - 1 = 32767$. Der Clipper vergleicht diesen Wert mit dem Betrag des abgetasteten Audiosignals und entscheidet nach Gleichung (1) welcher Wert dem Audio-Codec zugeführt wird. Daraus ergibt sich die im Bild 1 zu sehende Kennlinie für den Clipper.

$$y(n) = \begin{cases} -FS & \text{für } x(n) < -FS \\ x(n) & \text{für } -FS \leq x(n) \leq FS \\ FS & \text{für } x(n) > FS \end{cases} \quad (1)$$

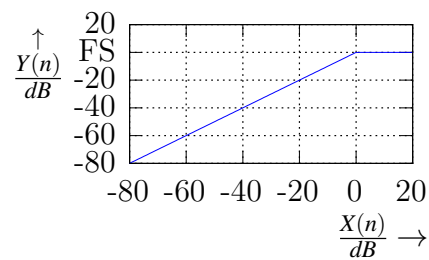


Bild 1: Kennlinie des Clippers

Bild 2 zeigt den Zeit- und Frequenzbereich bei dem der Clipper nicht begrenzt und Bild 3 bei dem er begrenzt. Im Spektrum des begrenzenden Clippers sind deutlich die hinzugekommenen Frequenzen zu sehen.

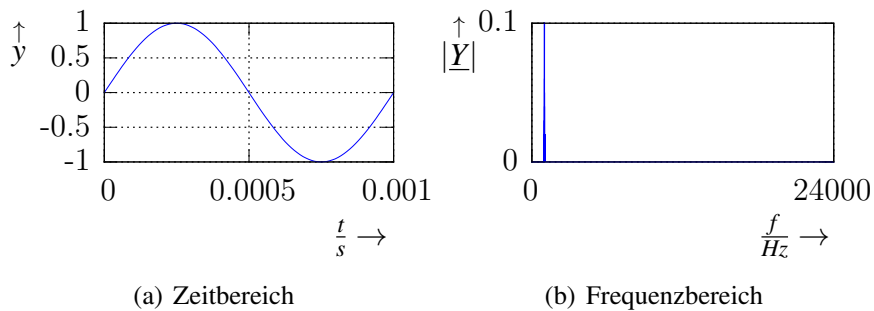


Bild 2: Signal ohne Begrenzung

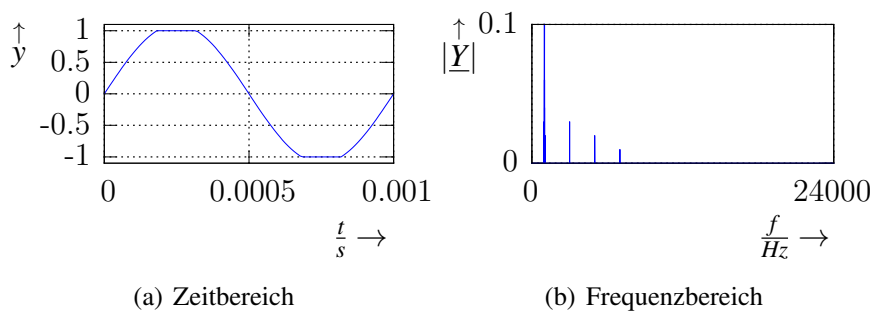


Bild 3: Signal mit Begrenzung

Störend werden die Verzerrungen wahrgenommen, wenn der Eingangspegel stark zunimmt und somit die Verzerrungen stärker ausfallen. Die stärker werdenden Verzerrungen haben den Ursprung in den weiter hinzukommenden Frequenzen. Auch erhöhen sich die Amplitudenwerte der hinzugekommenen Frequenzen mit zunehmenden Eingangspegel. Zum Vergleich zeigt Bild 4 den Zeit- und Frequenzbereich für ein Sinussignal mit einem Pegel von 0,83 dB. Full scale

wird hier mit 1 für lineare Werte und 0 dB für logarithmische Werte festgelegt. Im Bild 5 ist ein Sinussignal mit einem Pegel von 26 dB zu sehen. Die Unterschiede zwischen beiden Bildern sind nicht nur im Zeitbereich erkennbar, sondern vor allem im Frequenzbereich. Im Bild 4b sind kaum zusätzliche Frequenzen zu erkennen, wohingegen im Bild 5b neben der Grundfrequenz von 1 kHz sehr deutlich weitere Frequenzen zu sehen sind.

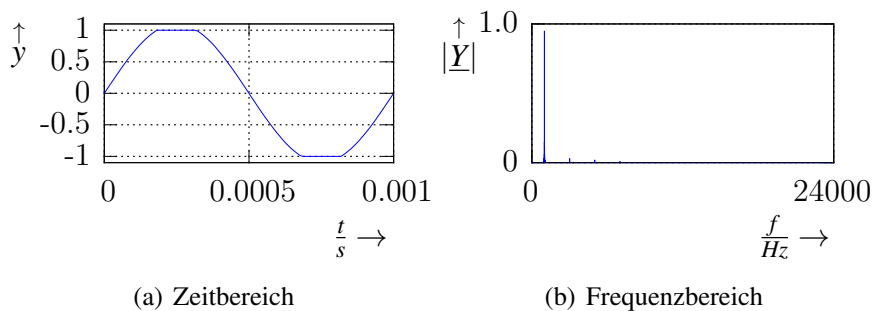


Bild 4: Signal schwach begrenzt

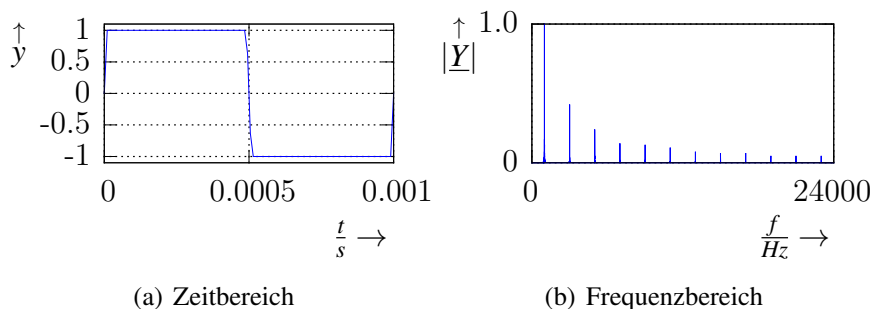


Bild 5: Signal stark begrenzt

Ein starkes Begrenzen mit der Folge starker Verzerrungen kann also durch die geeignete Wahl des Eingangspegels verhindert werden und der Clipper begrenzt tatsächlich nur die kurz auftretenden Spitzen, deren Verzerrungen kaum bis nicht hörbar sind, zumindest in den meisten Fällen als nicht störend wahrgenommen werden.

In Bild 6 wird nochmal das Signal aus Bild 4 mit einem Eingangspegel von 0,83 dB verwendet, welches dem Audio-Codec zugeführt wird ohne vorher den Clipper zu durchlaufen. Es kommt daher im Audio-Codec zu einem Überlauf dessen Auswirkung doch deutlich zu sehen sind. Es kann festgehalten werden, dass es bei einem Überlauf schon bei kleinen Pegeln zu starken hörbaren Verzerrungen kommt. Wohingegen es mit Verwendung des Clippers erst mit wesentlich höheren Pegeln zu hörbaren Verzerrungen kommt.

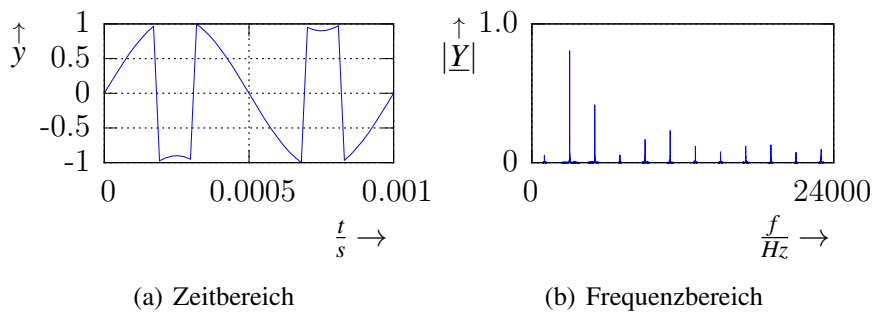


Bild 6: Signal wie in Bild 4 ohne Clipper

Weiter kann festgehalten werden, dass die Kombination aus Limiter und Clipper für nicht übermäßig stark verstärkte Ausgangssignale eine gute Kombination ist, um diese so zu begrenzen, dass es zu keinen hörbaren Verzerrungen kommt. Der Limiter begrenzt die Hüllkurve und der Clipper begrenzt kurze starke Impulse, welche vom Limiter nicht bzw. kaum begrenzt werden.

3.2. Der Echo-Effekt

Das Echo wurde der Klangregelung hinzugefügt, weil es ein einfach umzusetzender Effekt ist. Somit kann die einfache Erweiterbarkeit der Klangregelung gezeigt werden ohne sich zu sehr auf das hinzuzufügende Element konzentrieren zu müssen. Das Bild 7 zeigt den Signalflussgraphen des Echo-Effektes.

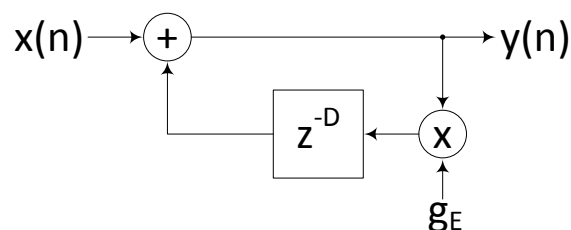


Bild 7: Signalflussgraph des Echo-Effektes

Es ist zu sehen, dass der Echo-Effekt eine rekursive Komponente besitzt. In dem rekursiven Zweig wird zunächst das Ausgangssignal $y(n)$ mit dem Verstärkungsfaktor g_E multipliziert. Das Ergebnis wird um D_E Abtastungen verzögert und anschließend dem Eingangssignal aufaddiert. Mit der Verzögerung D_E kann bestimmt werden, nach welcher Zeit t_E das gedämpfte Signal dem Eingangssignal hinzugefügt wird. Somit kann auch bestimmt werden, ob es als Hall ($t_E < 50 \text{ ms}$) oder als Echo ($t_E > 50 \text{ ms}$) wahrgenommen wird.

$$D_E = t_E \cdot f_s \quad (2)$$

Da das Ergebnis der Gleichung (2) für die Verzögerung $D_E \in \mathbb{N}$ sein muss, wird für die Abtastfrequenz $f_s \in \mathbb{N}$ und die Verzögerungszeit $t_E \in \mathbb{N}$ festgelegt. Somit wurde für die Einstellmöglichkeit der Verzögerungszeit t_E ein Bereich von 1 ms bis 1000 ms wählbar in 1-ms-Schritten festgelegt.

Mit dem Verstärkungsfaktor g_E kann bestimmt werden, wie stark das Signal gedämpft wird und somit wie oft bzw. wie lang es zu hören ist. Das Signal verschwindet nie, da das Signal nur gedämpft wird. Es ist nach einer gewissen Zeit nicht mehr wahrzunehmen, da das Signal so stark gedämpft ist, dass die Signalstärke nahezu 0 wird.

3.3. Der Dynamic Processor

Mit dem Echo-Effekt wurde ein einfacher Effekt hinzugefügt, um sich vorrangig mit dem Hinzufügen eines neuen Elementes zur Klangregelung zu beschäftigen. Beim Dynamic Processor geht es nun darum ein komplexeres Element dem Klangregler hinzuzufügen, um nachzuweisen, dass auch dies einfach zu realisieren ist. Dabei wird der Limiter entfernt und der Dynamic Processor hinzugefügt. Der Dynamic Processor beinhaltet vier Elemente, welche die Dynamik beeinflussen. Die vier Elemente sind ein Limiter, Compressor, Expander und ein Noise-Gate. Welches dieser vier Elemente zur Anwendung kommt, wird über einen Entscheider bestimmt. Zur Berechnung der Verstärkungsfaktoren der vier Elemente wird ein Peak- und ein Effektivwert-Detektor benötigt. Da der Rechenaufwand z.B. durch die Logarithmierung und Entlogarithmierung sehr hoch ist, wird eine Abtastratenreduktion dem Dynamic Processor hinzugefügt. Bild 8 zeigt den Signalflussgraphen des Dynamic Processors.

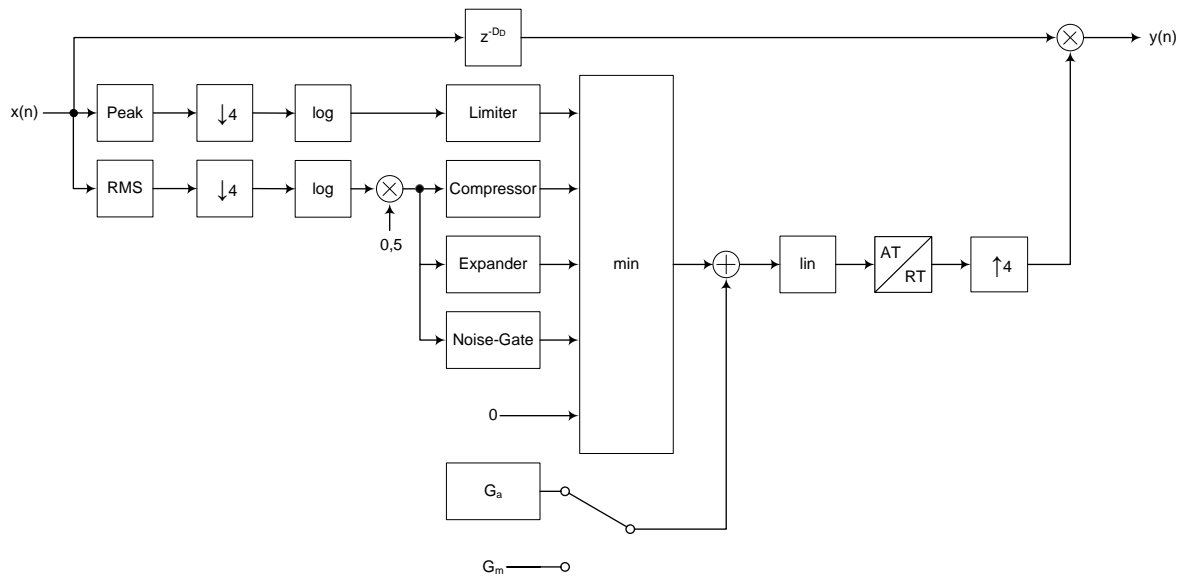


Bild 8: Signalflussgraph Dynamic Processor

3.3.1. Die Detektoren zur Ermittlung des Effektiv- und Spitzenwertes

Zur Berechnung der Verstärkungsfaktoren werden zwei Detektoren benötigt. Der erste Detektor ermittelt den Spitzenwert und der Zweite den Effektivwert. Der Limiter arbeitet mit dem Spitzenwert und der Compressor, der Expander und das Noise-Gate arbeiten mit dem Effektivwert.

Die Spitzenwertmessung erfolgt über eine Ansprech- t_a und Rücklaufzeit t_r . Aus diesen werden die Zeitkonstanten AT und RT gebildet, wie dies Abschnitt 3.3.3 entnommen werden kann. Zunächst werden die Eingangspegelwerte $x_{left}(n)$ für den linken Kanal und $x_{right}(n)$ für den rechten Kanal, gemäß Gleichungen (3) und (4), normalisiert.

$$x_l(n) = \frac{x_{left}(n)}{FS} \quad (3)$$

$$x_r(n) = \frac{x_{right}(n)}{FS} \quad (4)$$

Aus den normalisierten Eingangsepegeln wird ein additiver Wert $x(n)$ zur Bewertung ermittelt.

$$x(n) = x_l(n) + x_r(n) \quad (5)$$

Der Spitzenwert $x_{peak}(n)$ wird mit

$$x_{peak}(n) = \begin{cases} (1 - AT)x_{peak}(n-1) + AT \cdot |x(n)| & \text{für } |x(n)| > x_{peak}(n-1) \\ (1 - RT)x_{peak}(n-1) + RT \cdot |x(n)| & \text{für } |x(n)| \leq x_{peak}(n-1) \end{cases} \quad (6)$$

ermittelt. Für die Ermittlung des Effektivwertes wird der Mittelungskoeffizient TAV benötigt. Dieser ergibt sich, mit Gleichung (7), aus der Mittelungszeit t_M . Die Mittelungszeit t_M ist in Millisekunden anzugeben.

$$TAV = 1 - e^{-\frac{2200}{f_s t_M}} \quad (7)$$

Der Effektivwert $x_{RMS}(n)$ kann nun mit

$$x_{RMS}^2(n) = (1 - TAV)x_{RMS}^2(n-1) + TAV \cdot x^2(n) \quad (8)$$

ermittelt werden. Die Auflösung des Quadrates erfolgt nach der Logarithmierung, da dort keine Wurzel, sondern eine Division durch 2 erfolgt. Die Division kann, um die Geschwindigkeit der Berechnung zu erhöhen, als Multiplikation mit 0,5 erfolgen. Die Logarithmierung und das ziehen der Wurzel erfolgt nach Gleichung (10). Im weiteren Verlauf werden lineare Werte mit Kleinbuchstaben und deren logarithmische Werte mit Großbuchstaben angegeben.

$$X_{RMS}(n) = 20 \log_{10}(x_{RMS}^2(n)) \cdot 0,5 \quad (9)$$

$$X_{RMS}(n) = 10 \log_{10}(x_{RMS}^2(n)) \quad (10)$$

Dabei tritt ein Problem auf, wenn $x_{RMS}^2(n) = 0$ ist. Dies führt bei der Logarithmierung zu einem $X_{RMS} = NaN$. NaN bedeutet *Not a Number*. Dies liegt daran, dass der Logarithmus für $\log_{10}(x_{RMS}^2(n))$ für $x_{RMS}^2 \leq 0$ einen Grenzwert von

$$X_{RMS} = \lim_{x_{RMS}^2 \rightarrow 0} (10 \log_{10}(x_{RMS}^2)) \quad (11)$$

$$X_{RMS} = -\infty \quad (12)$$

hat. Da ein $X_{RMS} = -\infty$ in der Praxis nicht umsetzbar ist, wird für den Fall $x_{RMS}(n) = 0$ ein $X_{RMS}(n) = -10^{12}$ dB festgelegt. Somit erweitert sich Gleichung (10) zu Gleichung (13).

$$X_{RMS}(n) = \begin{cases} 10 \log_{10}(x_{RMS}^2(n)) & \text{für } x_{RMS}^2(n) > 0 \\ -10^{12} & \text{für } x_{RMS}^2(n) = 0 \end{cases} \quad (13)$$

Auch der Spitzenwert $x_{peak}(n)$ muss logarithmiert werden. Auch hier gelten die Gleichen Bedingungen wie bei der Logarithmierung des Effektivwertes, womit sich Gleichung (14) für die Logarithmierung des Spitzenwertes $x_{peak}(n)$ ergibt.

$$X_{peak}(n) = \begin{cases} 20 \log_{10}(x_{peak}(n)) & \text{für } x_{peak}(n) > 0 \\ -10^{12} & \text{für } x_{peak}(n) \leq 0 \end{cases} \quad (14)$$

3.3.2. Die Kennlinie des Dynamic Processors

Die Kennlinie des Dynamic Processors hängt von den gewählten Einstellungen des Anwenders für die Elemente Limiter, Compressor, Expander und Noise-Gate ab. Weiter hängt die Kennlinie davon ab, welche Komponenten aktiviert und deaktiviert sind, was auch für die automatische Verstärkung gilt.

Die vier verwendeten Elemente sind vom Aufbau gleich. Sie unterscheiden sich in ihrer Kennlinie, welche eine lineare Funktion ist, in der Steigung und Wirkungsrichtung ausgehend von der entsprechenden Schwelle. So kann die Funktion über die Steigung nach Tabelle 2, welche durch die Ratio (Kompressionsfaktor) repräsentiert wird, unterschieden werden.

Funktion	Ratio R
Limiter	$R = \infty$
Compressor	$R > 1$
Expander	$0 < R < 1$
Noise-Gate	$R = 0$

Tabelle 2: Zuordnung der Funktion zu den Kompressionsfaktoren (Ratio)

Für die Parameter der Funktionen werden im weiteren Verlauf folgende Symbole verwendet:

Element	Symbol				
	Enable/Disable	Threshold	Ratio	Slope	Soft Knee
Limiter	L_{en}	L_T	L_R	L_S	–
Compressor	C_{en}	C_T	C_R	C_S	C_K
Expander	E_{en}	E_T	E_R	E_S	E_K
Noise-Gate	N_{en}	N_T	N_R	N_S	–

Tabelle 3: Zuordnung der Symbole zu den Parametern der Funktionen des Dynamic Processors

Die Ratio gibt an wie groß die Änderung des Ausgangspegels in Bezug auf den Eingangspegel ist. So ergibt eine Ratio 4:1, für den Compressor, eine Pegeländerung am Ausgang um 1 dB, wenn am Eingang der Pegel sich um 4 dB geändert hat. Für den Expander werden Ratios kleiner 1 angegeben. Somit ergibt eine Ratio 1:4, für den Expander, eine Pegeländerung am Ausgang um 4 dB, wenn am Eingang eine Pegeländerung von 1 dB stattfand. In den Gleichungen zur Ermittlung der Verstärkungsfaktoren wird allerdings die Ratio nicht direkt verwendet, sondern der Slope (Steigung). Der Slope berechnet sich nach Gleichung (15)¹.

$$S = 1 - \frac{1}{R} \quad (15)$$

Der Limiter ist ein Spezialfall des Compressors, welcher eine starke Kompression aufweist. Diese ergibt sich aus der Ratio $L_R = \infty$. Dies ist ein theoretischer Wert, welcher sich in der Praxis nicht umsetzen lässt. Somit ist der Slope über eine Grenzwertbildung zu ermitteln.

$$L_S = \lim_{L_R \rightarrow \infty} \left(1 - \frac{1}{L_R} \right) \quad (16)$$

$$(17)$$

Somit ist für den Limiter Slope $L_S = 1$ zu verwenden.

¹[1], S. 233 Gl. (7.5)

Ähnlich sieht es mit der Ratio des Noise-Gate aus. Das Noise-Gate ist ein Spezialfall des Expanders mit einer starken Expansion. Die Ratio ist mit $N_R = 0$ festgelegt. Dies ergibt in Gleichung (15) eine Division durch Null, daher ist auch hier der Slope über eine Grenzwertbildung zu ermitteln.

$$N_S = \lim_{N_R \rightarrow 0} \left(1 - \frac{1}{N_R} \right) \quad (18)$$

Für den Noise-Gate Slope ergibt sich somit $N_S = -\infty$. Da dies sich in der Praxis nicht umsetzen lässt, wird das Noise-Gate Slope mit $N_S = -10^{12}$ festgelegt. Dies ist eine gute Näherung zum angestrebten Wert.

Die Schwellen (Thresholds) sind in dB anzugeben und stehen in Bezug zu Full scale FS = 0 dB. Über die Aktivierungssignale (X_{en}) kann ein Element des Dynamic Processors aktiviert oder deaktiviert werden. Die Signale können den Wert 1 für aktiviert (enabled) und den Wert 0 für deaktiviert (disabled) annehmen. Das Soft-Knee dient dazu die Schnittpunkte zwischen den Kennlinien abzurunden und den Übergang weicher zu gestalten. Somit kommt es an den Schnittstellen nicht zu einer sprunghaften Kompression bzw. Expansionen. Der Parameter C_K bzw. E_K ist in dB anzugeben. Damit wird angegeben wie weit vor bzw. nach dem entsprechenden Threshold der Übergang beginnen bzw. enden soll. Begonnen wird mit einer Ratio von 1:1 und am Ende wird die eingestellte Ratio eingenommen. Bevor hier die Berechnung des Soft-Knee hergeleitet wird, sollen zunächst erst einmal die Kennlinien der einzelnen Elemente ohne Soft-Knee hergeleitet werden, um bei der Herleitung des Soft-Knee ein besseres Verständnis zu erhalten.

Die Kennlinien der einzelnen Komponenten des Dynamic Processors sind lineare Funktionen, welche aus den logarithmierten Eingangspegel einen Verstärkungsfaktor berechnen, um damit einen bestimmten Ausgangspegel zu erreichen. Die Kennlinie in Bild 9 zeigt den normalen Verlauf zwischen Eingangs- und Ausgangspegel ohne Dynamic Processor. Es ist zu sehen, dass der Ausgangspegel Y dem Eingangspegel X entspricht.

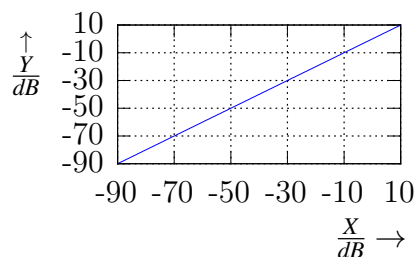


Bild 9: Kennlinie von Eingangspegel zu Ausgangspegel ohne Dynamic Processor

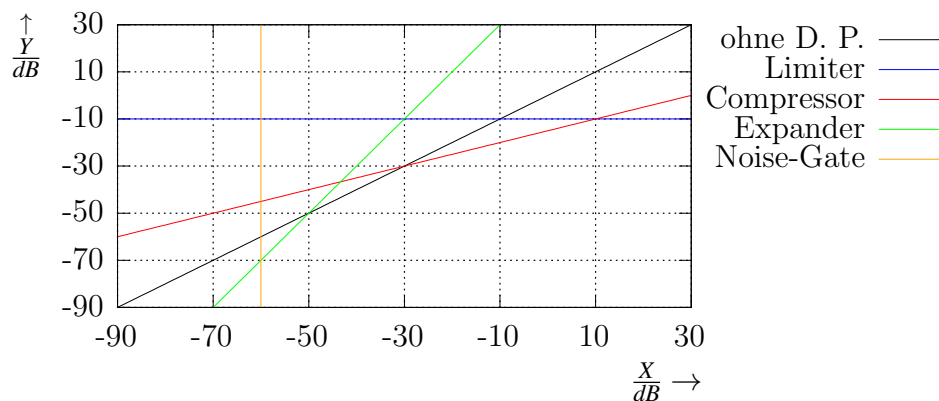


Bild 10: Kennlinie der einzelnen Elemente des Dynamic Processors

Bild 10 zeigt die Kennlinien der Komponenten des Dynamic Processors. Es wurden folgende Parameter gewählt:

$$\begin{aligned}
 L_T &= -10 \text{ dB} \\
 C_T &= -30 \text{ dB} & C_R &= 2 : 1 & C_S &= 0,5 \\
 E_T &= -50 \text{ dB} & E_R &= 1 : 2 & E_S &= -1,0 \\
 N_T &= -60 \text{ dB}
 \end{aligned}$$

Prinzipiell werden die Kennlinien nach Gleichung (19) berechnet. Dabei ist das Ergebnis der logarithmische Verstärkungsfaktor G der auf das Eingangssignal X anzuwenden ist, um die Kompression bzw. Expansion zu erreichen.

$$G(X) = S(T - X) \quad (19)$$

Aus dieser grundsätzlichen Gleichung ergeben sich für die Verstärkungsfaktoren der Komponenten des Dynamic Processors folgende Gleichungen:

$$L(X_{peak}) = L_S(L_T - X_{peak}) \quad (20)$$

$$C(X_{RMS}) = C_S(C_T - X_{RMS}) \quad (21)$$

$$E(X_{RMS}) = E_S(E_T - X_{RMS}) \quad (22)$$

$$N(X_{RMS}) = N_S(N_T - X_{RMS}) \quad (23)$$

Die Kennlinie des Dynamic Processors ergibt sich so aus den einzelnen Kennlinien der Komponenten anhand der eingestellten Thresholds. Für die Kompressoren kommt der Anteil der

Kennlinien oberhalb der Schwelle und für die Expander unterhalb der Schwelle zur Anwendung. Somit ergibt sich aus dem vorangegangenen Beispiel die Kennlinie wie sie Bild 11 zeigt.

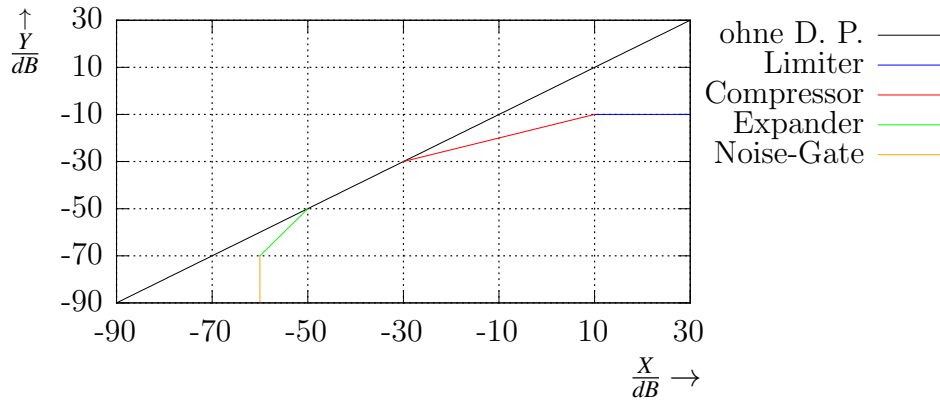


Bild 11: Kennlinie des Dynamic Processors

Die Kennlinie des Dynamic Processors ergibt sich daraus, dass der Verstärkungsfaktor des Dynamic Processors G_K sich aus dem kleinsten Verstärkungsfaktor der Komponenten ergibt.

$$G_K(L, C, E, N) = \min(0 \text{ dB}, N(X_{RMS}), E(X_{RMS}), C(X_{RMS}), L(X_{peak})) \quad (24)$$

Da sich die Kennlinie aus den Thresholds ergibt und diese auf das Eingangssignal bezogen sind, ist zu erkennen, dass die Kennlinie beim Übergang von Compressor zu Limiter nicht stimmt. Die Limiterschwelle ist auf $L_T = -10 \text{ dB}$ eingestellt. Der Limiter sollte somit bereits bei $X = -10 \text{ dB}$ arbeiten. Wie man aus Bild 11 erkennen kann beginnt der Limiter aber erst bei $X = 10 \text{ dB}$ zu wirken. Dies wird durch eine Anpassung der Gleichung (20) zu Gleichung (25) gelöst. In ähnlicher Weise wird auch Gleichung (23) zu Gleichung (26) geändert. Somit ergibt sich die Kennlinie des Dynamic Processors zu der in Bild 12 gezeigten.

$$L(X_{peak}) = L_S(L_T - X_{peak}) - C_S(L_T - C_T) \quad (25)$$

$$N(X_{RMS}) = N_S(N_T - X_{RMS}) + E_S(E_T - N_T) \quad (26)$$

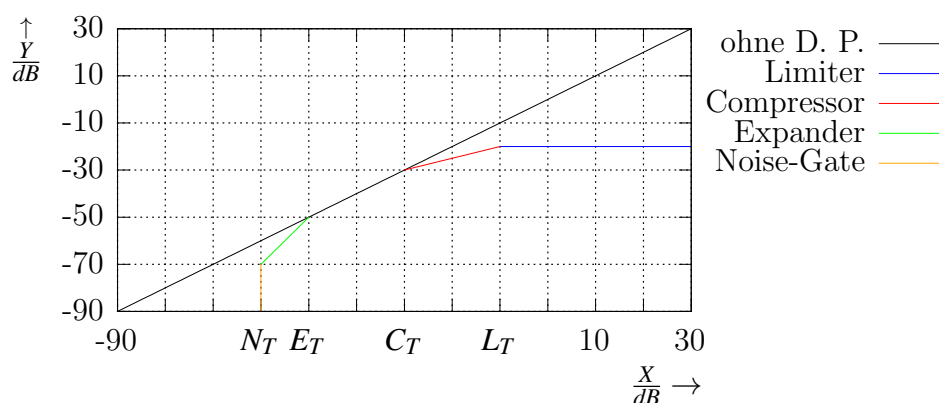


Bild 12: Kennlinie des Dynamic Processors mit angepassten Gleichungen

Mit den geänderten Gleichungen kann erkannt werden, dass der Limiter ab der eingestellten Schwelle $L_T = -10 \text{ dB}$ arbeitet. Es ist aber auch weiter zu sehen, dass der Ausgangspegel dadurch um weitere 10 dB niedriger ist. Ohne eine weitere Verstärkung nach der Kompression/Expansion ist der Ausgangspegel immer niedriger als der Eingangspegel. Somit ist eine Verstärkung nach der Kompression/Expansion notwendig, um den Effekt der Steigerung der Lautheit zu erzielen, denn dies ist der Zweck der Dynamikbeeinflussung. Dies wird durch die Komprimierung der lauten Passagen und die Expansion der leisen Passagen erreicht. Dadurch werden die Pegel zueinander angenähert und der Unterschied zwischen leisen und lauten Passagen wird verringert. Durch die ausgangsseitige Verstärkung (Hub) wird der Ausgangspegel angehoben und somit die Lautheit erhöht. Die ausgangsseitige Verstärkung kann durch einen manuell einzustellenden Verstärkungsfaktor G_m oder einen automatischen Verstärkungsfaktor G_a erreicht werden. Der manuelle Verstärkungsfaktor wird im folgenden als *Manual Output Gain* (MOG) und der automatische Verstärkungsfaktor mit *Auto Output Gain* (AOG) bezeichnet. Solch eine Dynamikbeeinflussung wird in der Regel bei lauten Umgebungen (Auto, Kaufhaus usw.) verwendet, damit das Audiosignal lauter erscheint und somit vom Hörer eher wahrgenommen wird als die Umgebung. Die Dynamikbeeinflussung kann aber auch zum halten eines konstanten Ausgangspegels verwendet werden (z. B. Rundfunk, Fernsehen usw.). Bild 13 zeigt die Wirkung des Kompressors und des Expanders auf ein Sinussignal.

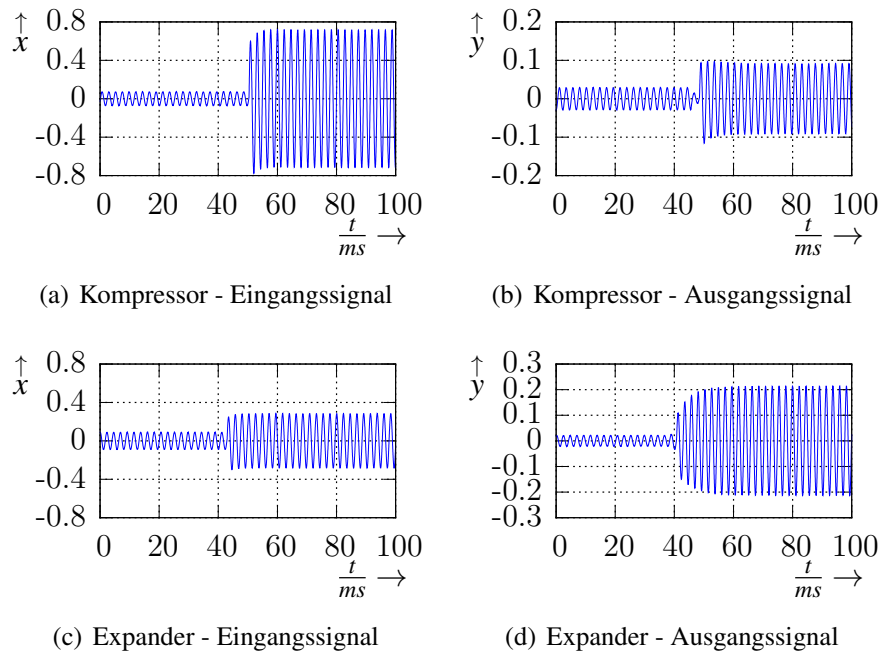


Bild 13: Wirkung von Kompressor und Expander

Bei der Berechnung des Verstärkungsfaktors G_a wird der Zustand des Limiters und des Compressors berücksichtigt. Dazu wird geprüft ob diese aktiviert oder deaktiviert sind. Je nach Zustand erfolgt die Berechnung nach folgenden Gleichungen. Dabei wird die Gleichung (25) als grundlegende Gleichung verwendet und bei deaktivierten Limiter oder Compressor die jeweilige Ratio auf 1 eingestellt. Somit ergibt sich ein Slope von 0 und das jeweilige Element ist aus der Gleichung eliminiert. Sind Limiter und Compressor aktiviert so ergibt sich mit den zuvor genannten Bedingungen Gleichung (30).

$$FS = L_S(L_T - X) - C_S(L_T - C_T) + G_a + X \quad (27)$$

$$0 \text{ dB} = L_S(L_T - 0) - C_S(L_T - C_T) + G_a + 0 \text{ dB} \quad (28)$$

$$0 \text{ dB} = L_S \cdot L_T - C_S(L_T - C_T) + G_a + 0 \text{ dB} \quad (29)$$

$$G_a = C_S(L_T - C_T) - L_S \cdot L_T \quad (30)$$

Gleichung (30) dient nun als weitere Grundlage für die Gleichungen bei deaktivierten Limiter oder Compressor. Mit $C_R = 1$ ergibt sich $C_S = 0$ und somit wird Gleichung (30) zu Gleichung (33) bei aktiviertem Limiter und deaktivierten Compressor.

$$G_a = C_S(L_T - C_T) - L_S \cdot L_T \quad (31)$$

$$G_a = 0 \text{ dB} \cdot (L_T - C_T) - L_S \cdot L_T \quad (32)$$

$$G_a = -L_S \cdot L_T \quad (33)$$

Ist der Compressor aktiviert und der Limiter deaktiviert, wird $L_R = 1$ gesetzt und somit $L_S = 0$. Eingesetzt in Gleichung (30) ergibt sich somit Gleichung (36)

$$G_a = C_S(L_T - C_T) - L_S \cdot L_T \quad (34)$$

$$G_a = C_S(L_T - C_T) - 0 \text{ dB} \cdot L_T \quad (35)$$

$$G_a = C_S(L_T - C_T) \quad (36)$$

Sind Limiter und Compressor deaktiviert, wird das Auto Output Gain deaktiviert und das Manual Output Gain aktiviert. Dies führt zu Gleichung (37).

$$G_a = G_m \quad (37)$$

Fasst man die Gleichungen (30), (33), (36) und (37) zu einer Gleichung zusammen, so ergibt sich daraus Gleichung (38) für das Auto Output Gain. Dabei sind die Variablen L_{en} und C_{en} logische Werte, welche den Zustand 0 (deaktiviert) oder 1 (aktiviert) annehmen können und repräsentieren den Aktivierungszustand des Limiters (L_{en}) und des Compressors (C_{en}).

$$G_a(L_{en}, C_{en}) = \begin{cases} C_S(L_T - C_T) - L_S \cdot L_T & \text{für } (L_{en} \wedge C_{en}) = 1 \\ -L_S \cdot L_T & \text{für } (L_{en} \wedge \neg C_{en}) = 1 \\ C_S(L_T - C_T) & \text{für } (\neg L_{en} \wedge C_{en}) = 1 \\ G_m & \text{für } (\neg L_{en} \wedge \neg C_{en}) = 1 \end{cases} \quad (38)$$

Für den Gesamtverstärkungsfaktor des Dynamic Processors $G_D(G_{en})$ wird nun der Verstärkungsfaktor $G_K(L, C, E, N)$ mit dem Verstärkungsfaktor $G_a(L_{en}, C_{en})$ oder G_m addiert. Ob der Verstärkungsfaktor $G_a(L_{en}, C_{en})$ des Auto Output Gain oder G_m des Manual Output Gain verwendet wird hängt vom Aktivierungszustand G_{en} ab. G_{en} ist ebenfalls ein logischer Wert, welcher den Zustand 0 für deaktiviert und 1 für aktiviert annehmen kann und steht dafür ob das

Auto Output Gain vom Anwender aktiviert oder deaktiviert wurde (siehe Gleichung (39)).

$$G_D(G_{en}) = G_K(L, C, E, N) + \begin{cases} G_a(L_{en}, C_{en}) & \text{für } G_{en} = 1 \\ G_m & \text{für } G_{en} = 0 \end{cases} \quad (39)$$

Da der Übergang zwischen den Kennlinien des Limiters und Compressors sowie des Noise-Gates und des Expanders sehr hart sein kann bzw. sprunghaft erfolgt, wird für diese Übergänge ein sogenanntes Soft-Knee eingeführt. Dieses soll den Übergang weicher gestalten. Dazu kommen die Variablen C_K für den Kompressor und E_K für den Expander zum Einsatz. Diese geben die Breite des Soft-Knee in dB an. Von der eingestellten Schwelle des Compressors/Expanders werden, gemäß eingestellter Soft-Knee-Breite C_K bzw. E_K , zwei Punkte oberhalb und Unterhalb entsprechend folgender Gleichungen festgelegt.

$$P_{C_0} = \left(C_T - \frac{C_K}{2}, C_T - \frac{C_K}{2} \right) \quad (40)$$

$$P_{C_2} = \left(C_T + \frac{C_K}{2}, C_T + \frac{C_K}{2} + C_S \left(C_T - \left(C_T + \frac{C_K}{2} \right) \right) \right) \quad (41)$$

$$P_{E_0} = \left(E_T + \frac{E_K}{2}, E_T + \frac{E_K}{2} \right) \quad (42)$$

$$P_{E_2} = \left(E_T - \frac{E_K}{2}, E_T - \frac{E_K}{2} + E_S \left(E_T - \left(E_T - \frac{E_K}{2} \right) \right) \right) \quad (43)$$

Diese Punkte sind die Start- und Endpunkte des jeweiligen Soft-Knee. Aus diesen Punkten lassen sich für die x- und y-Koordinatenwerte des Compressors die Gleichungen

$$C_{x_0} = C_T - \frac{C_K}{2} \quad (44)$$

$$C_{x_1} = C_T \quad (45)$$

$$C_{x_2} = C_T + \frac{C_K}{2} \quad (46)$$

$$C_{y_0} = C_{x_0} \quad (47)$$

$$C_{y_1} = C_{x_1} \quad (48)$$

$$C_{y_2} = C_{x_2} + C_S(C_T - C_{x_2}) \quad (49)$$

$$(50)$$

und für den Expander die Gleichungen

$$E_{x_0} = E_T + \frac{E_K}{2} \quad (51)$$

$$E_{x_1} = E_T \quad (52)$$

$$E_{x_2} = E_T - \frac{E_K}{2} \quad (53)$$

$$E_{y_0} = E_{x_0} \quad (54)$$

$$E_{y_1} = E_{x_1} \quad (55)$$

$$E_{y_2} = E_{x_2} + E_S(E_T - E_{x_2}) \quad (56)$$

$$(57)$$

extrahieren. Die x- und y-Werte für die Punkt P_{C_1} und P_{E_1} ergeben sich aus den Schwellen C_T und E_T . Diese sind die Wendepunkte in der zur Anwendung kommenden Bernstein-Bézier-Darstellung von Kurven und Flächen.

Die BERNSTEIN-BÉZIER-Darstellung (kurz B-B-Darstellung) von Kurven und Flächen verwendet die BERNSTEINSchen Grundpolynome

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad 0 \leq t \leq 1 \quad (i = 0, 1, \dots, n) \quad (58)$$

und nutzt vor allem die folgenden Eigenschaften aus:

$$1. \quad 0 \leq B_{i,n}(t) \leq 1 \quad \text{für } 0 \leq t \leq 1, \quad (59)$$

$$2. \quad \sum_{i=0}^n B_{i,n}(t) = 1.^2 \quad (60)$$

Zur B-B-Kurvendarstellung seien $n+1$ Eckpunkte $P_i (i = 0, 1, \dots, n)$ mit den Ortsvektoren \vec{P}_i eines räumlichen Polygons, das in einem Zusammenhang als Stützpolygon bezeichnet wird gegeben.

²[2], S. 1006 f.

Durch die Vorschrift

$$\vec{r}(t) = \sum_{i=0}^n B_{i,n}(t) \vec{P}_i, \quad 0 \leq t \leq 1, \quad (61)$$

wird diesen Punkten eine Raumkurve, die sogenannte B-B-Kurve zugeordnet. Wegen (60) kann (61) als „variable Konvexkombination“ der gegebenen Punkte aufgefaßt werden. Die Raumkurve (61) hat folgende wichtige Eigenschaften:

1. Die Punkte P_0 und P_n werden interpoliert.
2. Die Vektoren $\overrightarrow{P_0P_1}$ und $\overrightarrow{P_{n-1}P_n}$ sind Tangenten von $\vec{r}(t)$ in P_0 bzw. P_n .

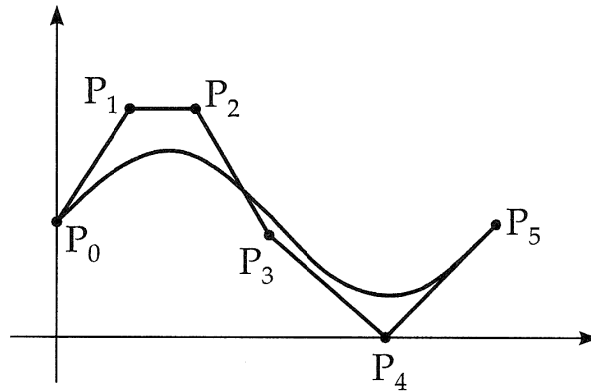


Bild 14: B-B-Kurve³

Den Zusammenhang zwischen Stützpolygon und B-B-Kurve zeigt Bild 14⁴.

Setzt man in Gleichung (58) $n = 2$ so erhält man folgende Polynome:

$$B_{0,2}(t) = \binom{2}{0} t^0 (1-t)^{2-0} = (1-t)^2 \quad (62)$$

$$B_{1,2}(t) = \binom{2}{1} t^1 (1-t)^{2-1} = 2t(1-t) \quad (63)$$

$$B_{2,2}(t) = \binom{2}{2} t^2 (1-t)^{2-2} = t^2 \quad (64)$$

Für die Punkte P_{C_i} und P_{E_i} ergeben sich für die x- und y-Werte aus Gleichung (61) die Folgenden:

$$x(t) = \sum_{i=0}^n B_{i,n}(t) x_i \quad (65)$$

$$y(t) = \sum_{i=0}^n B_{i,n}(t) y_i \quad (66)$$

³[2], S. 1007, Abb. 19.14

⁴[2], S. 1006 f.

Setzt man auch hier $n = 2$ so ergeben sich die Gleichungen (67) und (68).

$$x(t) = x_0(1-t)^2 + 2x_1t(1-t) + x_2t^2 \quad (67)$$

$$y(t) = y_0(1-t)^2 + 2y_1t(1-t) + y_2t^2 \quad (68)$$

Setzt man $x_0 = C_{x_0}$, $x_1 = C_{x_1}$ und $x_2 = C_{x_2}$ in Gleichung (67) ein erhält man den x-Koordinatenwert des Compressors C_x in Abhängigkeit der Variablen t. Stellt man die Gleichung (70) nach t um so ergibt sich Gleichung (72), welche eine Abhängigkeit zum Effektivwert $X_{RMS} = C_x$ hat.

$$C_x(t) = \left(C_T - \frac{C_K}{2}\right)(1-t)^2 + 2C_Tt(1-t) + \left(C_T + \frac{C_K}{2}\right)t^2 \quad (69)$$

$$C_x(t) = C_T - \frac{C_K}{2} + C_Kt \quad (70)$$

$$t = \frac{C_x - C_T + \frac{C_K}{2}}{C_K} \quad (71)$$

$$t(X_{RMS}) = C_{c_0}X_{RMS} + C_{c_1} \quad \text{mit } C_{c_0} = \frac{1}{C_K}; \quad C_{c_1} = \frac{1}{2} - \frac{C_T}{C_K}; \quad X_{RMS} = C_x \quad (72)$$

Setzt man $y_0 = C_{y_0}$, $y_1 = C_{y_1}$ und $y_2 = C_{y_2}$ in Gleichung (68) ein erhält man den y-Koordinatenwert des Compressors C_y in Abhängigkeit der Variablen t.

$$C_y(t) = C_{x_0}(1-t)^2 + 2C_{x_1}t(1-t) + [C_{x_2} + C_S(C_T - C_{x_2})]t^2 \quad (73)$$

$$C_y(t) = \left(C_T - \frac{C_K}{2}\right)(1-t)^2 + 2C_Tt(1-t) + \left\{\left(C_T + \frac{C_K}{2}\right) + C_S\left[C_T - \left(C_T + \frac{C_K}{2}\right)\right]\right\}t^2 \quad (74)$$

$$C_y(t) = C_T - \frac{C_K}{2} + C_Kt - \frac{C_S C_K}{2}t^2 \quad (75)$$

$$C_y(t) = C_{c_2} + C_Kt + C_{c_3}t^2 \quad \text{mit } C_{c_2} = C_T - \frac{C_K}{2}; \quad C_{c_3} = -\frac{C_S C_K}{2} \quad (76)$$

Setzt man in diese Gleichung t , aus Gleichung (72), ein erhält man Gleichung (78), welche den Ausgangswert des Compressors in Abhängigkeit zum Effektivwert X_{RMS} darstellt.

$$C_y(X_{RMS}) = C_{c_2} + C_K(C_{c_0}X_{RMS} + C_{c_1}) + C_{c_3}(C_{c_0}X_{RMS} + C_{c_1})^2 \quad (77)$$

$$C_y(X_{RMS}) = C_{c_4} + C_{c_5}X_{RMS} + C_{c_6}X_{RMS}^2 \quad \text{mit } C_{c_4} = C_{c_2} + C_{c_1}C_K + C_{c_3}C_{c_1}^2 \quad (78)$$

$$C_{c_5} = C_{c_0}C_K + 2C_{c_0}C_{c_1}C_{c_3}$$

$$C_{c_6} = C_{c_3}C_{c_0}^2$$

Genauso ist mit den Expanderpunkten vorzugehen. Dazu sind $x_0 = E_{x_0}$, $x_1 = E_{x_1}$ und $x_2 = E_{x_2}$ in Gleichung (67) einzusetzen und nach t umzustellen.

$$E_x(t) = \left(E_T + \frac{E_K}{2}\right)(1-t)^2 + 2E_Tt(1-t) + \left(E_T - \frac{E_K}{2}\right)t^2 \quad (79)$$

$$E_x(t) = E_T + \frac{E_K}{2} - E_Kt \quad (80)$$

$$t = \frac{E_T + \frac{E_K}{2} - E_x}{E_K} \quad (81)$$

$$t = E_{c_0} + E_{c_1}E_x \quad \text{mit } E_{c_0} = 0,5 + \frac{E_T}{E_K}; \quad E_{c_1} = -\frac{1}{E_K} \quad (82)$$

Setzt man auch hier $y_0 = E_{y_0}$, $y_1 = E_{y_1}$ und $y_2 = E_{y_2}$ in Gleichung (68) ein erhält man Gleichung (86).

$$E_y(t) = E_{x_0}(1-t)^2 + 2E_{x_1}t(1-t) + [E_{x_2} + E_S(E_T - E_{x_2})]t^2 \quad (83)$$

$$E_y(t) = \left(E_T + \frac{E_K}{2}\right)(1-t)^2 + 2E_Tt(1-t) + \left\{\left(E_T - \frac{E_K}{2}\right) + E_S\left[E_T - \left(E_T - \frac{E_K}{2}\right)\right]\right\}t^2 \quad (84)$$

$$E_y(t) = E_T + \frac{E_K}{2} - E_Kt + \frac{E_S E_K}{2}t^2 \quad (85)$$

$$E_y(t) = E_{c_2} - E_Kt + E_{c_3}t^2 \quad \text{mit } E_{c_2} = E_T + \frac{E_K}{2}; \quad E_{c_3} = \frac{E_S E_K}{2} \quad (86)$$

Setzt man nun t in Gleichung (86) ein, so erhält man auch hier den Ausgangswert des Expanders in Abhängigkeit des Effektivwertes X_{RMS} .

$$E_y(X_{RMS}) = E_{c_2} - E_K(E_{c_0} + E_{c_1}X_{RMS}) + E_{c_3}(E_{c_0} + E_{c_1}X_{RMS})^2 \quad (87)$$

$$E_y(X_{RMS}) = E_{c_4} + E_{c_5}X_{RMS} + E_{c_6}X_{RMS}^2 \quad \text{mit } E_{c_4} = E_{c_2} - E_{c_0}E_K + E_{c_3}E_{c_0}^2 \quad (88)$$

$$E_{c_5} = 2E_{c_0}E_{c_1}E_{c_3} - E_{c_1}E_K$$

$$E_{c_6} = E_{c_3}E_{c_1}^2 \quad (89)$$

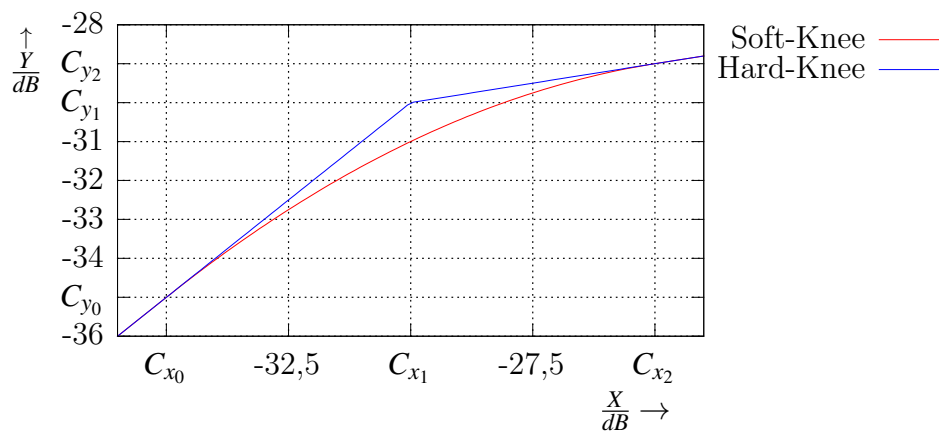
Die Konstanten $C_{c_0}, C_{c_1}, C_{c_2}, C_{c_3}, C_{c_4}, C_{c_5}, C_{c_6}, E_{c_0}, E_{c_1}, E_{c_2}, E_{c_3}, E_{c_4}, E_{c_5}$ und E_{c_6} sind zur Minimierung des Rechenaufwands eingeführt wurden. Diese Konstanten müssen nur bei entsprechenden Parameteränderungen, durch den Anwender, neu berechnet werden. Im weiteren Verlauf werden diese weiter zusammengefasst, um eine weitere Minimierung des Rechenaufwandes zu erreichen.

Die Gleichungen (21) und (22) werden um das Soft-Knee zu

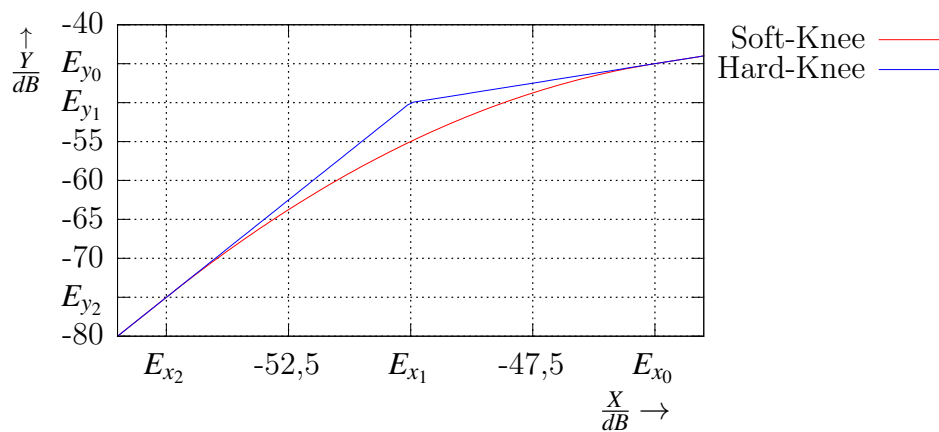
$$C(X_{RMS}) = \begin{cases} C_y(X_{RMS}) - X_{RMS} & \text{für } C_{x_0} < X_{RMS} < C_{x_2} \\ C_S(C_T - RMS) & \text{sonst} \end{cases} \quad (90)$$

$$E(X_{RMS}) = \begin{cases} E_y(X_{RMS}) - X_{RMS} & \text{für } E_{x_2} < X_{RMS} < E_{x_0} \\ E_S(C_T - X_{RMS}) & \text{sonst} \end{cases} \quad (91)$$

erweitert. Bild 15 zeigt ein Beispiel für das Soft-Knee des Compressors und Expanders mit einer Soft-Knee-Breite $C_K = 10 \text{ dB}$ und $E_K = 10 \text{ dB}$, sowie einer Ratio $C_R = 5$ und $E_R = 0,2$.



(a) Compressor



(b) Expander

Bild 15: Soft-Knee - Compressor/Expander

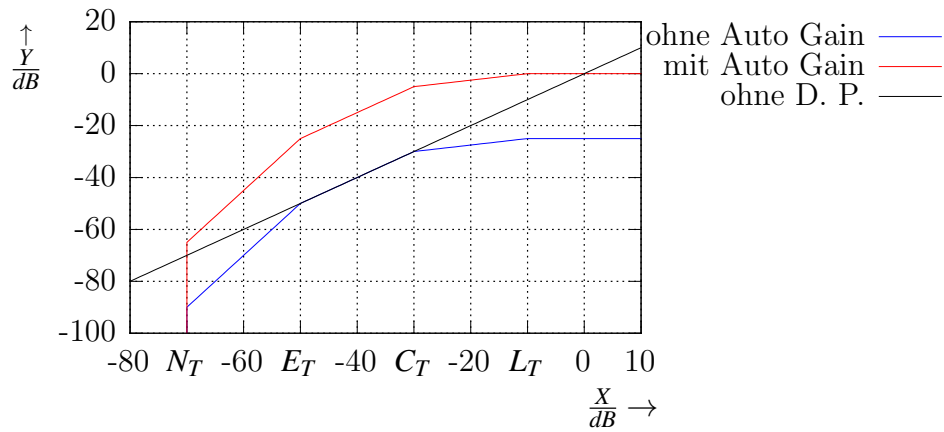
Bild 16 zeigt die gesamte Kennlinie des Dynamic Processors mit deaktivierten und aktivierten Soft-Knee. Zusätzlich wird in den Diagrammen zwei Kurven angezeigt. Diese unterscheiden sich darin, dass das Auto Output Gain aktiviert bzw. deaktiviert ist. Bei den Beispielen in den Bildern wurden folgende Einstellungen vorgenommen:

$$L_T = -10.0 \text{ dB}$$

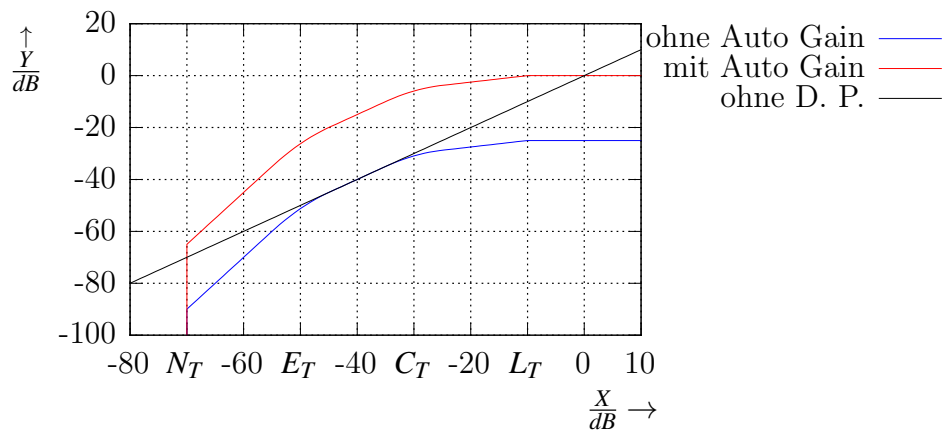
$$C_T = -30.0 \text{ dB} \quad C_R = 4 \quad C_K = 10.0 \text{ dB}$$

$$E_T = -50.0 \text{ dB} \quad E_R = 0,5 \quad E_K = 10.0 \text{ dB}$$

$$N_T = -70.0 \text{ dB}$$



(a) mit Hard-Knee



(b) mit Soft-Knee

Bild 16: Dynamic Processor Kennlinie (vollständig)

Da die Variablen C_{c_0} , C_{c_1} , C_{c_2} , C_{c_3} , E_{c_0} , E_{c_1} , E_{c_2} und E_{c_3} nur Zwischenergebnisse zur Berechnung der benötigten Variablen C_{c_4} , C_{c_5} , C_{c_6} , E_{c_4} , E_{c_5} und E_{c_6} sind, liegt hier noch Potential um die Berechnung der Variablen zu vereinfachen und die Variablen für die Zwischenergebnisse entfallen zu lassen, um somit den Rechenaufwand weiter zu minimieren und Speicherplatz einzusparen. Die Gleichungen

$$C_{c_4} = C_{c_2} + C_{c_1} C_K + C_{c_3} C_{c_1}^2 \quad (92)$$

$$C_{c_5} = C_{c_0} C_K + 2C_{c_0} C_{c_1} C_{c_3} \quad (93)$$

$$C_{c_6} = C_{c_3} C_{c_0}^2 \quad (94)$$

lassen sich durch Einsetzen der Variablen C_{c_0} , C_{c_1} , C_{c_2} und C_{c_3} zu

$$C_{c_4} = \frac{C_S}{2} \left(C_T - \frac{C_K}{4} - \frac{C_T^2}{C_K} \right) \quad (95)$$

$$C_{c_5} = 1 + C_S \left(\frac{C_T}{C_K} - \frac{1}{2} \right) \quad (96)$$

$$C_{c_6} = -\frac{C_S}{2C_K} \quad (97)$$

minimieren. Auf die gleiche Weise lassen sich die Gleichungen

$$E_{c_4} = E_{c_2} - E_{c_0}E_K + E_{c_3}E_{c_0}^2 \quad (98)$$

$$E_{c_5} = 2E_{c_0}E_{c_1}E_{c_3} - E_{c_1}E_K \quad (99)$$

$$E_{c_6} = E_{c_3}E_{c_1}^2 \quad (100)$$

durch Einsetzen der Variablen E_{c_0} , E_{c_1} , E_{c_2} und E_{c_3} zu

$$E_{c_4} = \frac{E_S}{2} \left(E_T + \frac{E_K}{4} + \frac{E_T^2}{E_K} \right) \quad (101)$$

$$E_{c_5} = 1 - E_S \left(\frac{1}{2} + \frac{E_T}{E_K} \right) \quad (102)$$

$$E_{c_6} = \frac{E_S}{2E_K} \quad (103)$$

minimieren.

3.3.3. Die Ansprech- und Rücklaufzeit

Die Ansprech- und Rücklaufzeit des Dynamic Processors wird auf den ermittelten Verstärkungsfaktor $G_D(G_{en})$ des Dynamic Processors angewandt. Dazu ist der logarithmische Verstärkungsfaktor $G_D(G_{en})$ in die Folge $f(n)$ nach Gleichung (104) umzuwandeln.

$$f(n) = 10^{\frac{G_D(G_{en})}{20}} \quad (104)$$

Der Steuerfaktor $f(n)$ wird nun mit dem vorherigen linearen Verstärkungsfaktor $g_D(n-1)$ verglichen. Gemäß Gleichung (105) wird anhand des Vergleiches der neue Verstärkungsfaktor $g_D(n)$ ermittelt.

$$g_D(n) = \begin{cases} (1-AT)g_D(n-1) + AT \cdot f(n) & \text{für } f(n) < g_D(n-1) \\ (1-RT)g_D(n-1) + RT \cdot f(n) & \text{für } f(n) \geq g_D(n-1) \end{cases} \quad (105)$$

Der so ermittelte Verstärkungsfaktor wird nun auf das verzögerte Eingangssignal multipliziert und somit der neue Ausgangswert des Dynamic Processors ermittelt.

$$y(n) = x(n - D_D) \cdot g_D(n) \quad (106)$$

Die Koeffizienten AT und RT können aus der eingestellten Ansprech- t_a und Rücklaufzeit t_r entsprechend den Gleichungen (107) und (108) berechnet werden. Die Ansprech- t_a und Rücklaufzeit t_r sind in Millisekunden anzugeben.

$$AT = 1 - e^{-\frac{2200}{fs t_a}} \quad (107)$$

$$RT = 1 - e^{-\frac{2200}{fs t_r}} \quad (108)$$

Eine genaue Herleitung findet sich in [1], S. 234 ff.

3.3.4. Abtastratenreduktion

Nach der Detektion des Spitzen- und Effektivwertes erfolgt eine Unterabtastung, um den Rechenaufwand zu minimieren. Da die Signale $X_{peak}(n)$ und $X_{RMS}(n)$ schon bandbegrenzt sind, kann die Unterabtastung direkt ausgeführt werden, indem nur jeder zweite oder jeder vierte Wert diesen beiden Folgen entnommen wird. Hiermit kann die Logarithmierung (LD(x)), Kennlinienberechnung (KENN), Entlogarithmierung (2^X) und Filterung für Ansprech- und Rücklaufzeit (SMO) mit reduzierter Abtastrate durchgeführt werden. Die danach folgende Aufwärtstastung um den Faktor 4 wird durch vierfaches Wiederholen des Ausgangswertes des Zeitkonstantenfilters erreicht. Dieser Vorgang entspricht der Aufwärtstastung um den Faktor 4 und anschließender Filterung mit der Abtast-Halte-Übertragungsfunktion.⁵

Die Verschachtelung und Verteilung von Teilprogrammen über vier Abtasttakte hinweg ist in Bild 17 dargestellt. Die Programmteile Peak/RMS (Spitzenwert-/Effektivwertberechnung) und MULT (Verzögerung des Eingangssignals und Multiplikation mit $g_D(n)$) werden in jedem Abtastintervall ausgeführt. Die Anzahl der Prozessorzyklen für Peak/RMS ist mit Z1 bezeichnet und die für MULT mit Z3. Die Programmteile LD(x), KENN, 2^X und SMO haben die maximale Prozessorzyklenanzahl Z2 und werden in den vier aufeinanderfolgenden Abtastintervallen nacheinander abgearbeitet. Dieser Vorgang wiederholt sich alle vier Abtasttakte. Die Gesamtanzahl von Prozessorzyklen pro Abtasttakt für den vollständigen Dynamik-Algorithmus ergibt sich aus der Summe aller drei Teilprogramme.⁶

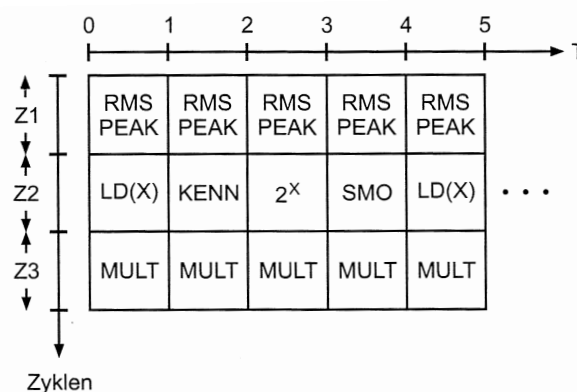


Bild 17: Verschachtelungstechnik⁷

⁵[1], S. 241 f.

⁶[1], S. 242 f.

⁷[1], S. 243, Bild 7.14

4. Die Software

In den folgenden Abschnitten wird dargestellt, wie die Erweiterungen Clipper, Echo und Dynamic Processor in die Bestandteile der Software der Klangregelung eingearbeitet wurden.

4.1. Das Audio Signal Processing Program (ASPP)

In der Audiosignalverarbeitung sind der Clipper und der Echo-Effekt hinzugekommen. Der Limiter wurde entfernt und durch den Dynamic Processor ersetzt. Dazu wurde die Datei *limiter.h* und *limiter.c* entfernt. Hinzugekommen sind für den Clipper die Dateien *clipper.h* und *clipper.c*, für das Echo die Dateien *echo.h* und *echo.c* und für den Dynamic Processor die Dateien *dynamic.h* und *dynamic.c*. Es ist zu sehen, dass für jeden Block in der Audiosignalverarbeitung ein eigener Programmteil, bestehend aus Header- (xxx.h) und Programmdatei (xxx.c), zur Anwendung kommt. Weiter eingeführt wurde die Datei *defines.h* für globale Definitionen.

In der Header-Datei (xxx.h) sind weitere benötigte Header-Dateien einzubinden, Definitionen zu erstellen und die globalen Variablen und die Funktionen für den entsprechenden Audio-signalverarbeitungsblock zu deklarieren. Wie dies aussehen kann zeigt das Listing der Datei *echo.h*.

Listing 1: echo.h (V.1.3)

```
20 #ifndef _ASPP_ECHO_H
21 #define _ASPP_ECHO_H
22
23
24 // Header-Dateien einbinden
25 #include <cs1.h>
26
27
28
29
30 // Minimal- und Maximalwerte definieren
31 #define min_E_tE 1 // minimalste Echozeit
32 #define max_E_tE 1000 // maximalste Echozeit
33 #define min_E_GE -20 // minimalster Verstärkungsfaktor
34 #define max_E_GE -1 // maximalster Verstärkungsfaktor
35
36 #define max_E_buf_size 48000 // fs * max_tE
37
38
39
40 // Standardwerte
41 #define default_E_en FALSE // Ein-/Ausschalten Echo
42 #define default_E_tE 250 // Verzögerungszeit in ms
43 #define default_E_GE -9 // Verstärkungsfaktor in dB
```

```

44
46
48 // globale Variablen deklarieren
49 Bool   E_en;           // Echo enable/disable
50 Uint16 E_tE;           // Echozeit in Millisekunden
51 Uint16 E_buf_size;     // Echozeit in Samples
52 Int8   E_GE;           // logarithmischer Verstärkungsfaktor Echo
53 float  E_gE;           // linearer Verstärkungsfaktor Echo
54 Uint32 E_i;            // Speicherplatzindex (Echo Index)
55 float  E_buf_r[max_E_buf_size]; // Puffer für rechten Kanal
56 float  E_buf_l[max_E_buf_size]; // Puffer für linker Kanal
57
58 // Puffervariablen zum SDRAM routen
59 #pragma DATA_SECTION(E_buf_r, ".EXT_RAM")
60 #pragma DATA_SECTION(E_buf_l, ".EXT_RAM")
61
62
63 // Funktionen deklarieren
64 void Echo_Reset_Buffer();
65 void Echo_Set_Delay(Uint16 delay);
66 void Echo_Set_Amp(Int8 G);
67 void Echo_Init();
68 void Echo(float *sample_left, float *sample_right);
69
70
71
72
73
74 #endif

```

Anschließend ist in der Programmdatei (xxx.c) der Quellcode zu den Funktionen zu hinterlegen. Typischerweise gibt es mindestens eine Initialisierungs- und eine Ausführungsfunktion, wie man es im folgenden Listing der Datei *echo.c* sehen kann. Die Initialisierungsfunktion ist im Funktionsnamen durch das *Init* gekennzeichnet und der Bezeichner der Ausführungsfunktion entspricht der Bezeichnung des Audiosignalverarbeitungsblocks. Im Fall des Echo-Effektes wurde so die Ausführungsfunktion mit *Echo* bezeichnet. Alle anderen Funktionen sollten nach dem Prinzip $\langle \text{Blockbezeichnung} \rangle_ \langle \text{Funktionsbezeichnung} \rangle$ benannt werden, so dass die Funktion dem jeweiligen Block und der Funktionalität eindeutig zugeordnet werden kann.

Listing 2: echo.c (V.1.3)

```

20 #ifndef _ASPP_ECHO_C
21 #define _ASPP_ECHO_C
22
23
24 // Header-Dateien einbinden
25 #include "echo.h" // eigene Header-Datei
26 #include <math.h> // mathematische Funktionen
27 #include <stdio.h>
28 #include <c67fastMath.h>
29
30
31
32
33 /*-----**
34 ** Die Funktion Echo_Reset_Buffer setzt den **
35 ** Puffer für den Echo-Effekt zurück. **
36 **-----**
37 void Echo_Reset_Buffer()
38 {
39     int i;
40     for(i=0; i<E_buf_size; i++)
41     {

```

```

42     E_buf_l[i] = 0.0;
43     E_buf_r[i] = 0.0;
44 }
45     E_i = 0;
46 }
47
48
49
50 /*-----**
51 ** Mit der Funktion Echo_set_delay wird die Verzögerungszeit eingestellt. **
52 ** Der Parameter delay steht für die Verzögerungszeit und diese ist in **
53 ** Millisekunden zu übergeben. **
54 **-----**/
55 void Echo_Set_Delay(UINT16 delay)
56 {
57     extern float fs;
58     E_tE = delay;
59     if (delay > max_E_tE) E_tE = max_E_tE;
60     E_buf_size = (fs * E_tE) * 0.001;
61     Echo_Reset_Buffer();
62 }
63
64
65
66 /*-----**
67 ** Mit der Funktion Echo_set_Amp wird der Verstärkungsfaktor **
68 ** eingestellt. Dieser ist im Parameter G in dB zu übergeben. **
69 **-----**/
70 void Echo_Set_Amp(INT8 G)
71 {
72     E_gE = G;
73     if (G > max_E_gE) E_gE = max_E_gE;
74     else if (G < min_E_gE) E_gE = min_E_gE;
75     E_gE = powsp(10.0, ((float)E_gE*0.05));
76 }
77
78
79
80 /*-----**
81 ** Die Funktion Echo_Init wird von der Funktion Init aufgerufen, **
82 ** um den Echo-Effekt zu initialisieren. **
83 **-----**/
84 void Echo_Init()
85 {
86     E_en = default_E_en;
87     Echo_Set_Delay(default_E_tE);
88     Echo_Set_Amp(default_E_gE);
89 }
90
91
92
93
94 /*-----**
95 ** Die Funktion Echo repräsentiert den Echo-Effekt. **
96 **-----**/
97 void Echo(float *sample_left, float *sample_right)
98 {
99     if (E_en)
100     {
101         (*sample_right) += E_buf_r[E_i];
102         (*sample_left) += E_buf_l[E_i];
103         E_buf_r[E_i] = (*sample_right) * E_gE;
104         E_buf_l[E_i] = (*sample_left) * E_gE;
105     } else {
106         E_buf_r[E_i] = (*sample_right);
107         E_buf_l[E_i] = (*sample_left);
108     }
109
110     E_i++;
111     if (E_i >= E_buf_size) E_i = 0;
112 }
113
114
115
116 #endif

```

Die Ausführungsfunktion muss die Samples für den linken und rechten Kanal mittels *Call by reference* übergeben bekommen (siehe Listing 2 Zeile 96). Somit müssen nicht für jeden Block extra Kopien der Samples erstellt werden. Diese werden so direkt bearbeitet. Die Ausführungsdateien sind entsprechend der Ausführungsreihenfolge in der Interrupt Service Routine *audio_isr* in der Datei *audio.c* einzubinden (siehe Listing 3 Zeilen 212 bis 216).

Listing 3: audio.c (V.1.3)

```

202 interrupt void audio_isr()
203 {
204     Uint32 sample;
205
206     // Daten abholen
207     sample = MCBSP_read(DSK6713_AIC23_DATAHANDLE);
208     sample_left = (float)(Int16)(Uint16)(sample >> 16);
209     sample_right = (float)(Int16)(Uint16)(sample & 0x0000FFFF);
210
211     // Signalverarbeitungskette
212     Filter(&sample_left, &sample_right);
213     Echo(&sample_left, &sample_right);
214     Audio(&sample_left, &sample_right);
215     Dynamic(&sample_left, &sample_right);
216     Clipper(&sample_left, &sample_right);
217
218     // Daten übergeben
219     sample = (Uint16)sample_left << 16;
220     sample = sample | (Uint16)sample_right;
221     MCBSP_write(DSK6713_AIC23_DATAHANDLE, sample);
222 }

```

Die Initialisierungsfunktionen, wie die *Echo_Init*, sollen die jeweiligen Audiosignalverarbeitungsblöcke initialisieren. Sie werden in die Funktion *Init* in der Datei *main.c* eingebunden (siehe Listing 4).

Listing 4: main.c (V.1.3)

```

32 void Init ()
33 {
34     Uint8 i;
35
36     IRQ_globalDisable ();
37
38     for (i=0; i<32; i++)
39     {
40         IRQ_disable (i);
41         IRQ_clear (i);
42     }
43
44     Filter_Init ();
45     Echo_Init ();
46     Dynamic_Init ();
47     Clipper_Init ();
48     i2c_Init ();
49     Audio_Init ();
50
51     IRQ_globalEnable ();
52 }

```

Damit die Kommunikation zwischen AU und CU stattfinden kann, müssen die Dateien *i2c.h* und *i2c.c* angepasst werden.

Zunächst sind für die Parameter des jeweiligen Blockes Kommandos zu definieren. Diese Kommandowerte stellen die Registeradresse zum jeweiligen Parameter dar (siehe Listing 5).

Listing 5: i2c.h (V.1.3)

```

52 // Kommandos definieren
#define cmd_input_vol 0x00 // Eingangslautstärke AIC23
54 #define cmd_deemph 0x01 // De-emphasis control (disabled, 32 kHz, 44.1 kHz, 48 kHz)
#define cmd_dac 0x02 // DAC select (off/selected)
56 #define cmd_byp 0x03 // Bypass on/off (AIC23 - Bypass)
#define cmd_lp_en 0x04 // Tiefpass ein/aus
58 #define cmd_lp_l_en 0x05 // Enable/Disable Tiefpass für linken Kanal
#define cmd_lp_r_en 0x06 // Enable/Disable Tiefpass für rechten Kanal
60 #define cmd_lp_fc_high 0x07 // Grenzfrequenz Tiefpass - High-Byte
#define cmd_lp_fc_low 0x08 // Grenzfrequenz Tiefpass - Low-Byte
62 #define cmd_lp_G 0x09 // logarithmischer Verstärkungsfaktor Tiefpass
#define cmd_hp_en 0x0A // Hochpass ein/aus
64 #define cmd_hp_l_en 0x0B // Enable/Disable Hochpass für linken Kanal
#define cmd_hp_r_en 0x0C // Enable/Disable Hochpass für rechten Kanal
66 #define cmd_hp_fc_high 0x0D // Grenzfrequenz Hochpass - High-Byte
#define cmd_hp_fc_low 0x0E // Grenzfrequenz Hochpass - Low-Byte
68 #define cmd_hp_G 0x0F // logarithmischer Verstärkungsfaktor Hochpass
#define cmd_vol 0x10 // Volume
70 #define cmd_mute_time 0x11 // Ein-/Ausblendzeit beim Mute
#define cmd_balance 0x12 // Balancereglerwert
72 #define cmd_mute 0x13 // Mute
#define cmd_D_en 0x14 // Ein-/Ausschalten des Dynamic Processors
74 #define cmd_D_l_en 0x15 // Ein-/Ausschalten des Dynamic Processors für den linken Kanal
#define cmd_D_r_en 0x16 // Ein-/Ausschalten des Dynamic Processors für den rechten Kanal
76 #define cmd_D_L_en 0x17 // Ein-/Ausschalten des Limiters
#define cmd_D_C_en 0x18 // Ein-/Ausschalten des Compressors
78 #define cmd_D_E_en 0x19 // Ein-/Ausschalten des Expanders
#define cmd_D_N_en 0x1A // Ein-/Ausschalten des Noise-Gates
80 #define cmd_D_G_en 0x1B // Ein-/Ausschalten des Auto Output Gains
#define cmd_D_LT 0x1C // Limiter Threshold
82 #define cmd_D_CT 0x1D // Compressor Threshold
#define cmd_D_ET 0x1E // Expander Threshold
84 #define cmd_D_NT 0x1F // Noise-Gate Threshold
#define cmd_D_CR 0x20 // Compressor Ratio
86 #define cmd_D_ER 0x21 // Expander Ratio
#define cmd_D_CK 0x22 // Compressor Soft-Knee
88 #define cmd_D_EK 0x23 // Expander Soft-Knee
#define cmd_D_G_m 0x24 // manueller Ausgangsverstärkungsfaktor
90 #define cmd_D_ta_high 0x25 // Ansprechzeit High-Byte
#define cmd_D_ta_low 0x26 // Ansprechzeit Low-Byte
92 #define cmd_D_tr_high 0x27 // Rücklaufzeit High-Byte
#define cmd_D_tr_low 0x28 // Rücklaufzeit Low-Byte
94 #define cmd_E_en 0x29 // Echo - Enable/Disable
#define cmd_E_GE 0x2A // Echo - logarithmischer Verstärkungsfaktor
96 #define cmd_E_tE_high 0x2B // Echo - Verzögerungszeit in Millisekunden (oberes Byte)
#define cmd_E_tE_low 0x2C // Echo - Verzögerungszeit in Millisekunden (oberes Byte)
98 #define cmd_clipper_en 0x2D // Ein-/Ausschalten Clipper

```

Bei der Übertragung über den I²C-Bus wird zunächst das Kommando (Adresse) gesendet und anschließend der Wert auf den der über den Kommandowert angesprochenen Parameter geändert werden soll. Damit der Parameter auch geändert wird, muss in der Datei *i2c.c* die Funktion *i2c_check* angepasst werden. Dazu ist in der switch-case-Anweisung das neue Kommando mit den entsprechenden Anweisungen zur Parameteränderung hinzuzufügen. Die Änderung sollte über eine eigene Anweisung erfolgen, wie es im Listing 6 für den Parameter *E_GE* zu sehen ist. Hier wird der Wert für den logarithmischen Verstärkungsfaktor G_E des Echos über die Funktion *Echo_Set_Amp* geändert. Der Funktion wird der Wert übergeben. Der übergebene Wert sollte über die Funktion validiert werden und erst nach der Validierung der

Parameter entsprechend auf diesen Wert geändert werden.

Listing 6: i2c.c (V.1.3)

```
246     case cmd_E_tE : Echo_Set_Amp((Int8)code);  
                    break;
```

Für 16-bit-Werte sind zwei Kommandowerte zu erstellen. Das Erste für das obere Byte und der Zweite für das untere Byte. Ebenso sind in der case-switch-Anweisung zwei case-Anweisungen zu hinterlegen. Die Erste reagiert auf das Kommando des oberen Bytes und übernimmt dieses und das Zweite reagiert auf das Kommando für das untere Byte und veranlasst die Zusammenführung der Beiden zu einem 16-bit-Wort. Im Listing 7 ist diese Verfahrensweise für den Parameter E_tE zu sehen.

Listing 7: i2c.c (V.1.3)

```
250     case cmd_E_tE_high : high = code;  
                    break;  
252     case cmd_E_tE_low  : low = code;  
                    valU16 = high;  
254                    valU16 = valU16 << 8;  
                    valU16 = valU16 | low;  
256                    Echo_Set_Delay(valU16);  
                    break;
```

4.2. Das Control and User Interface Program (CUIP)

Die Änderungen im CUIP belaufen sich hauptsächlich darauf, die neuen Parameter hinzuzufügen und die Parameter des Limiters zu entfernen. Des Weiteren sind die Bildschirme des LCD anzupassen.

Zunächst war die Konfiguration entsprechend zu ändern. Dazu wurde die Struktur *dsp_config* in der Datei *cuip_dsp.h* angepasst. Vergleiche dazu Listing 8 (Version 1.0) mit Listing 9 (Version 1.3).

Listing 8: cuip_dsp.h (V.1.0)

```
106 struct dsp_config {  
107     int s_G_ADC; // Code des AIC23 zur Einstellung der Eingangslautstärke  
108     int s_d; // De-Emphasis enabled/disabled  
109     int s_DAC_en; // DAC select  
110     int s_b; // Bypass  
111     int s_lp_en; // Ein-/Ausschalten Tiefpass  
112     int s_lp_l_en; // Ein-/Ausschalten Tiefpass linker Kanal
```

```

114 int s_lp_r_en; // Ein-/Ausschalten Tiefpass rechter Kanal
int s_lp_fc; // Tiefpass – Grenzfrequenz in Hz
int s_lp_G; // logarithmischer Verstärkungsfaktor des Tiefpasses in dB
116 int s_hp_en; // Ein-/Ausschalten Hochpass
int s_hp_l_en; // Ein-/Ausschalten Hochpass linker Kanal
118 int s_hp_r_en; // Ein-/Ausschalten Hochpass rechter Kanal
int s_hp_fc; // Hochpass – Grenzfrequenz
120 int s_hp_G; // logarithmischer Verstärkungsfaktor des Hochpasses in dB
int s_G_V; // logarithmischer Verstärkungsfaktor fürs Volume
122 int s_t_F; // Mute Ein-/Ausblendzeit
int s_B; // Balance (rechter – linker – Kanal)
124 int s_m; // Ein-/Ausschalten Mute
int s_lt_en; // Ein-/Ausschalten Limiter
126 int s_lt; // Threshold in %
int s_ta; // Attack Time in µs
128 int s_tr; // Release Time in ms
};

```

Listing 9: cuip_dsp.h (V.1.3)

```

172 struct dsp_config {
int s_G_ADC; // Code des AIC23 zur Einstellung der Eingangslautstärke
174 int s_d; // De-Emphasis enabled/disabled
int s_DAC_en; // DAC select
176 int s_b; // Bypass
int s_lp_en; // Ein-/Ausschalten Tiefpass
178 int s_lp_l_en; // Ein-/Ausschalten Tiefpass linker Kanal
int s_lp_r_en; // Ein-/Ausschalten Tiefpass rechter Kanal
180 int s_lp_fc; // Tiefpass – Grenzfrequenz in Hz
int s_lp_G; // logarithmischer Verstärkungsfaktor des Tiefpasses in dB
182 int s_hp_en; // Ein-/Ausschalten Hochpass
int s_hp_l_en; // Ein-/Ausschalten Hochpass linker Kanal
184 int s_hp_r_en; // Ein-/Ausschalten Hochpass rechter Kanal
int s_hp_fc; // Hochpass – Grenzfrequenz
186 int s_hp_G; // logarithmischer Verstärkungsfaktor des Hochpasses in dB
int s_G_V; // logarithmischer Verstärkungsfaktor fürs Volume
188 int s_t_F; // Mute Ein-/Ausblendzeit
int s_B; // Balance (rechter – linker – Kanal)
190 int s_m; // Ein-/Ausschalten Mute
int s_D_en; // Enable/Disable Dynamic Processor
192 int s_D_l_en; // Enable/Disable Dynamic Processor linker Kanal
int s_D_r_en; // Enable/Disable Dynamic Processor rechter Kanal
194 int s_D_L_en; // Enable/Disable Limiter
int s_D_C_en; // Enable/Disable Compressor
196 int s_D_E_en; // Enable/Disable Expander
int s_D_N_en; // Enable/Disable Noise-Gate
198 int s_D_G_en; // Enable/Disable Auto Output Gain
int s_D_LT; // Limiter Threshold
200 int s_D_CT; // Compressor Threshold
int s_D_ET; // Expander Threshold
202 int s_D_NT; // Noise-Gate Threshold
int s_D_CR; // Compressor Ratio
204 int s_D_ER; // Expander Ratio
int s_D_CK; // Breite des Compressor Soft-Knee
206 int s_D_EK; // Breite des Expander Soft-Knee
int s_D_G_m; // logarithmischer Verstärkungsfaktor des Manual Output Gain
208 int s_D_ta; // Ansprechzeit
int s_D_tr; // Rücklaufzeit
210 int s_E_en; // Echo Enable/Disable
int s_E_GE; // Echo – logarithmischer Verstärkungsfaktor
212 int s_E_tE; // Echo – Verzögerungszeit in Millisekunden
int s_clipper_en; // Enable/Disable Clipper
214 };

```

Entsprechend wurden auch die Definitionen für die Standardwerte, die Minimal- und Maximalwerte der Parameter in der Datei *cuip_dsp.h* angepasst bzw. hinzugefügt. Auch die Strukturen *dsp_commands* und *dsp_keys* müssen ebenfalls angepasst werden.

Der nächste Schritt ist die Änderung der Datei *cuip_dsp.cpp*. Hier muss im Konstruktor die Zuweisung der Standardwerte an die Konfiguration angepasst werden. Vergleiche dazu Listing 10 mit Listing 11.

Listing 10: *cuip_dsp.cpp* (V.1.0)

```

CUIP_DSP::CUIP_DSP(void)
32 {
    // Standardkonfiguration initialisieren
34     config = {
        default_s_G_ADC , // Code für AIC23 zur Einstellung der Eingangslautstärke (23 entspricht 0dB)
36         default_s_d , // De-emphases (0 – disabled , 1 – 32 kHz, 2 – 44,1 kHz, 3 – 48 kHz)
        default_s_DAC_en , // DAC select (0 – disabled , 1 – enabled)
38         default_s_b , // Bypass (0 – disabled , 1 – enabled)
        default_s_lp_en , // Enable/Disable Tiefpass
40         default_s_lp_l_en , // Enable/Disable Tiefpass linker Kanal
        default_s_lp_r_en , // Enable/Disable Tiefpass rechter Kanal
42         default_s_lp_fc , // Grenzfrequenz Tiefpass
        default_s_lp_G , // logarithmischer Verstärkungsfaktor Tiefpass in dB
44         default_s_hp_en , // Enable/Disable Hochpass
        default_s_hp_l_en , // Enable/Disable Hochpass linker Kanal
46         default_s_hp_r_en , // Enable/Disable Hochpass rechter Kanal
        default_s_hp_fc , // Grenzfrequenz Hochpass
48         default_s_hp_G , // logarithmischer Verstärkungsfaktor Hochpass in dB
        default_s_G_V , // logarithmischer Verstärkungsfaktor Ausgangslautstärke in dB
50         default_s_t_F , // Ein-/Ausblendzeit der Audio beim Ein- bzw. Auschalten des Mute
        default_s_B , // Balance – Reglerwert
52         default_s_m , // Enable/Disable Mute
        default_s_lt_en , // Enable/Disable Limiter
54         default_s_lt , // Limiter – Threshold in %
        default_s_ta , // Limiter – Attack Time in µs
56         default_s_tr // Limiter – Release Time in ms
    };
58     init_commands(); // Kommandos initialisieren
    init_keys(); // Schlüsselwörter initialisieren
60 }

```

Listing 11: *cuip_dsp.cpp* (V.1.3)

```

CUIP_DSP::CUIP_DSP(void)
32 {
    // Standardkonfiguration initialisieren
34     config = {
        default_s_G_ADC , // Code für AIC23 zur Einstellung der Eingangslautstärke (23 entspricht 0dB)
36         default_s_d , // De-emphases (0 – disabled , 1 – 32 kHz, 2 – 44,1 kHz, 3 – 48 kHz)
        default_s_DAC_en , // DAC select (0 – disabled , 1 – enabled)
38         default_s_b , // Bypass (0 – disabled , 1 – enabled)
        default_s_lp_en , // Enable/Disable Tiefpass
40         default_s_lp_l_en , // Enable/Disable Tiefpass linker Kanal
        default_s_lp_r_en , // Enable/Disable Tiefpass rechter Kanal
42         default_s_lp_fc , // Grenzfrequenz Tiefpass
        default_s_lp_G , // logarithmischer Verstärkungsfaktor Tiefpass in dB
44         default_s_hp_en , // Enable/Disable Hochpass
        default_s_hp_l_en , // Enable/Disable Hochpass linker Kanal
46         default_s_hp_r_en , // Enable/Disable Hochpass rechter Kanal
        default_s_hp_fc , // Grenzfrequenz Hochpass
48         default_s_hp_G , // logarithmischer Verstärkungsfaktor Hochpass in dB
        default_s_G_V , // logarithmischer Verstärkungsfaktor Ausgangslautstärke in dB
50         default_s_t_F , // Ein-/Ausblendzeit der Audio beim Ein- bzw. Auschalten des Mute
        default_s_B , // Balance – Reglerwert
52         default_s_m , // Enable/Disable Mute
        default_s_D_en , // Enable/Disable Dynamic Processor
54         default_s_D_l_en , // Enable/Disable Dynamic Processor linker Kanal
        default_s_D_r_en , // Enable/Disable Dynamic Processor rechter Kanal
56         default_s_D_L_en , // Enable/Disable Limiter
        default_s_D_C_en , // Enable/Disable Compressor
58         default_s_D_E_en , // Enable/Disable Expander
        default_s_D_N_en , // Enable/Disable Noise-Gate
60         default_s_D_G_en , // Enable/Disable Auto Output Gain
        default_s_D_LT , // Limiter Threshold
62         default_s_D_CT , // Compressor Threshold
        default_s_D_ET , // Expander Threshold
    };

```

```

64     default_s_D_NT ,      // Noise-Gate Threshold
    default_s_D_CR ,      // Compressor Ratio
66     default_s_D_ER ,      // Expander Ratio
    default_s_D_CK ,      // Breite des Compressor Soft-Knee in dB
68     default_s_D_EK ,      // Breite des Expander Soft-Knee in dB
    default_s_D_G_m ,     // logarithmischer Verstärkungsfaktor des Manual Output Gain
70     default_s_D_ta ,     // Ansprechzeit
    default_s_D_tr ,     // Rücklaufzeit
72     default_s_E_en ,     // Echo - Enable/Disable
    default_s_E_GE ,     // Echo - logarithmischer Verstärkungsfaktor
74     default_s_E_tE ,     // Echo - Verzögerungszeit in ms
    default_s_clipper_en // Enable/Disable Clipper
76 };
CR = config.s_D_CR;      // aktuellen Wert als alten Wert für die Compressor Ratio merken
78 ER = config.s_D_ER;    // aktuellen Wert als alten Wert für die Expander Ratio merken
G_en = config.s_D_G_en; // aktuellen Wert als alten Wert für das Enable/Disable Signal des Auto output Gain
80 init_commands();      // Kommandos initialisieren
init_keys();            // Schlüsselwörter initialisieren
82 }

```

Weiterhin werden die Attribute CR und ER als Zwischenspeicher eingeführt. Wird der Compressor oder der Expander deaktiviert, wird hier der alte Zustand hinterlegt, um diesen bei Aktivieren wieder herstellen zu können. Beim Deaktivieren des Compressors/Expanders wird die Ratio auf 1:1 eingestellt, damit dieselben Gleichungen verwendet werden können und nicht zusätzlich Quellcode für eine Fallunterscheidung notwendig wird. Ähnliches gilt für das Attribut G_en. Mit diesem wird hinterlegt, ob das AOG aktiviert war oder nicht, wenn der Limiter und der Compressor deaktiviert wird. Werden Compressor und Limiter deaktiviert, wird auch das AOG deaktiviert.

Auch die Methoden *init_dsp*, *set*, *get*, *add*, *sub*, *set_to_factory_settings*, *init_commands*, *init_keys* und *check_config* müssen an die hinzugekommenen und entfernten Parameter angepasst werden. Für die Methode *check_config* bzw. zur Validierung der Parameter müssen entsprechende check-Methoden hinzugefügt werden. Diese müssen den bereits vorhandenen entsprechen. Hierzu ein Beispiel der check-Methode zur Validierung des Parameters *s_lp_fc*:

Listing 12: cuip_dsp.cpp (V.1.3)

```
void CUIP_DSP::check_s_lp_fc(int *val) { check_config_value(val, max_s_lp_fc, min_s_lp_fc); }
```

Zusätzlich wurden für den Dynamic Processor die Methoden *set_L_en*, *set_LT*, *set_C_en*, *set_CR*, *set_E_en*, *set_ER* und *set_G_en* hinzugefügt. Deren Funktion kann den Kommentaren des Quellcodes entnommen werden.

Der nächste Schritt ist die Anpassung des Menüs, welches im LCD angezeigt wird. Dieses wird in den Dateien *cuip_gpio.h* und *cuip_gpio.cpp* verwaltet. In der Datei *cuip_gpio.h* müssen die Definitionen zum Menü entsprechend Listing 13 angepasst werden.

Listing 13: cuip_gpio.h (V.1.3)

```

62 // Menü – Konstanten
#define menu_level_counts 2 // Gibt die Anzahl der Menüebenen an
64 #define menu_main_items 9 // Anzahl Untermenüs im Hauptmenü (Exit das kein Untemenü ist, ist mit eingerechnet)

66 // Anzahl der Menüpunkte in einem Untermenü
#define menu_lowpass_items 5 // Lowpass
68 #define menu_highpass_items 5 // Highpass
#define menu_echo_items 3 // Echo
70 #define menu_dynamic_items 19 // Dynamic
#define menu_audio_items 5 // Audio
72 #define menu_audiocodec_items 2 // Audio-Codec
#define menu_network_items 2 // Network
74 #define menu_options_items 5 // Options
#define menu_exit_items 1 // Exit

76 // Menü – Schlüsselwerte
78 #define menu_off -1
#define menu_on 0
80 #define menu_lowpass 0
#define menu_highpass 1
82 #define menu_echo 2
#define menu_dynamic 3
84 #define menu_audio 4
#define menu_audiocodec 5
86 #define menu_network 6
#define menu_options 7
88 #define menu_exit 8
#define menu_lp_en 0
90 #define menu_lp_l_en 1
#define menu_lp_r_en 2
92 #define menu_lp_fc 3
#define menu_lp_G 4
94 #define menu_hp_en 0
#define menu_hp_l_en 1
96 #define menu_hp_r_en 2
#define menu_hp_fc 3
98 #define menu_hp_G 4
#define menu_E_en 0
100 #define menu_E_GE 1
#define menu_E_tE 2
102 #define menu_D_en 0
#define menu_D_l_en 1
104 #define menu_D_r_en 2
#define menu_D_L_en 3
106 #define menu_D_C_en 4
#define menu_D_E_en 5
108 #define menu_D_N_en 6
#define menu_D_G_en 7
110 #define menu_D_LT 8
#define menu_D_CT 9
112 #define menu_D_ET 10
#define menu_D_NT 11
114 #define menu_D_CR 12
#define menu_D_ER 13
116 #define menu_D_CK 14
#define menu_D_EK 15
118 #define menu_D_G_m 16
#define menu_D_ta 17
120 #define menu_D_tr 18
#define menu_audio_vol 0
122 #define menu_audio_mute 1
#define menu_audio_fade 2
124 #define menu_audio_balance 3
#define menu_audio_clipper 4
126 #define menu_ac_in_vol 0
#define menu_ac_deemph 1
128 #define menu_network_mode 0
#define menu_network_ip 1
130 #define menu_network_mask 2
#define menu_network_gw 3
132 #define menu_network_dns 4
#define menu_option_lcd_time 0
134 #define menu_option_load 1
#define menu_option_save 2
136 #define menu_option_reset 3

```

```
#define menu_factory_settings 4
```

In der Datei *cuip_gpio.cpp* sind die Methoden *Key_2_Pin_A_Change*, *Key_3_Pin_A_Change*, *Key_3_Pin_D_Falling*, *Key_4_Pin_A_Change*, *Key_5_Pin_A_Change*, *lcd_menu_change* und *lcd_menu_change_level_1* anzupassen. Zu den Anpassungen wurden die Methoden *Key_2_Pin_A_Change_Echo*, *Key_2_Pin_A_Change_Dynamic*, *Key_3_Pin_A_Change_Echo*, *Key_3_Pin_A_Change_Dynamic*, *Key_4_Pin_A_Change_Echo*, *Key_4_Pin_A_Change_Dynamic*, *Key_5_Pin_A_Change_Dynamic*, *lcd_menu_change_echo*, *getRatioString*, *lcd_menu_change_dynamic*, *set_lcd_main_dynamic* hinzugefügt. Deren Funktionsweise ist dem Quellcode des CUIP zu entnehmen. Das Aktivieren bzw. Deaktivieren des Clippers wurde dem Menü *Audio* hinzugefügt. Dazu mussten die Methoden *Key_2_Pin_A_Change_Audio* und *lcd_menu_change_audio* angepasst werden. Da der Limiter mit dem Dynamic Processor ausgetauscht wurde, wurden die Methoden *lcd_menu_change_limiter*, *Key_2_Pin_A_Change_Limiter*, *Key_3_Pin_A_Change_Limiter*, *Key_4_Pin_A_Change_Limiter*, *Key_5_Pin_A_Change_Limiter* und *set_lcd_main_lt* entfernt. Entsprechend ist natürlich die Datei *cuip_gpio.h* angepasst worden.

Weitere Änderungen erfolgten in den Methoden *load* und *save* der Klasse *CUIP_SD*, welche in den Dateien *cuip_sd.h* und *cuip_sd.cpp* hinterlegt sind. Hier mussten Anpassungen vorgenommen werden, dass die neuen Parameter auf die micro-SD card geschrieben und gelesen werden. Entsprechend waren die Parameter des Limiters zu entfernen und die des Clippers, Echos und Dynamic Processors hinzuzufügen.

Da das TCGUIP und TCWIP ebenfalls die Änderungen erhalten haben, waren auch die Klassen *CUIP_UDP* und *CUIP_WI* anzupassen. Die Anpassungen erfolgten in den Methoden *checkUDP*, *getConfigString*, *get_set_response* und *get_cfg_response*.

Der Hauptbildschirm des LCD wurde ebenfalls geändert. Hier wird an der dritten Position nicht mehr die Limiterschwelle des alten Limiters angezeigt, sondern die des Dynamic Processors. Mit dem Taster F3 wird nicht mehr der Limiter aktiviert/deaktiviert, sondern der gesamte Dynamic Processor. Zeigt der Hauptbildschirm an der dritten Position nicht mehr die Limiterschwelle, sondern die Zeichenkette „-“ ist damit nicht mehr gemeint, dass der Limiter deaktiviert ist, sondern es wird damit ausgedrückt, dass der Dynamic Processor deaktiviert wurde. Für diese Änderungen musste die Methode *print_main* der Klasse *CUIP_LCD*, welche in der Datei *cuip_lcd.cpp* hinterlegt ist, angepasst werden.

4.3. Das Tone Control Graphical User Interface Program (TCGUIP)

Die Änderungen am TCGUIP erfolgten hauptsächlich an der Benutzeroberfläche und in der Kommunikation mit der CU. Die Benutzeroberfläche wurde so abgeändert, dass das Panel des Limiters entfernt wurde und an der Stelle das Panel des Echos eingefügt wurde. Das Fenster wurde in der Höhe vergrößert, so dass die Einstellmöglichkeiten der Panels *Audio-Codec* und *Options* in zwei Zeilen angeordnet werden konnten. Somit konnte auf der rechten Seite Platz für das Panel *Dynamic Processor* und *Clipper* geschaffen werden. Das Panel des Clippers beinhaltet lediglich eine Checkbox zum Aktivieren/Deaktivieren des Clippers. Das Panel des Dynamic Processors beinhaltet Checkboxes zum Aktivieren/Deaktivieren des Dynamic Processors und des Dynamic processors left channel bypass bzw. right channel bypass. Weiter findet sich die Schaltfläche mit der Beschriftung *open parameter dialog*. Mit dieser Schaltfläche lässt sich ein weiteres Fenster öffnen in denen die Einstellungen für den Dynamic Processor vorgenommen werden können. Des Weiteren beinhaltet dieses Fenster ein Diagramm welches die Kennlinie des Dynamic Processors entsprechend seines Zustandes zeigt. Bild 18 zeigt das Fenster mit den Einstellmöglichkeiten für den Dynamic Processor und Bild 19 im Vergleich das Hauptfenster der Version 1.0 und der Version 1.3.

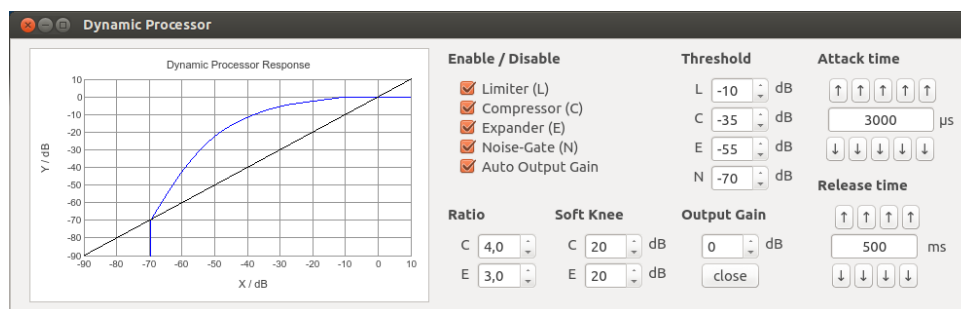
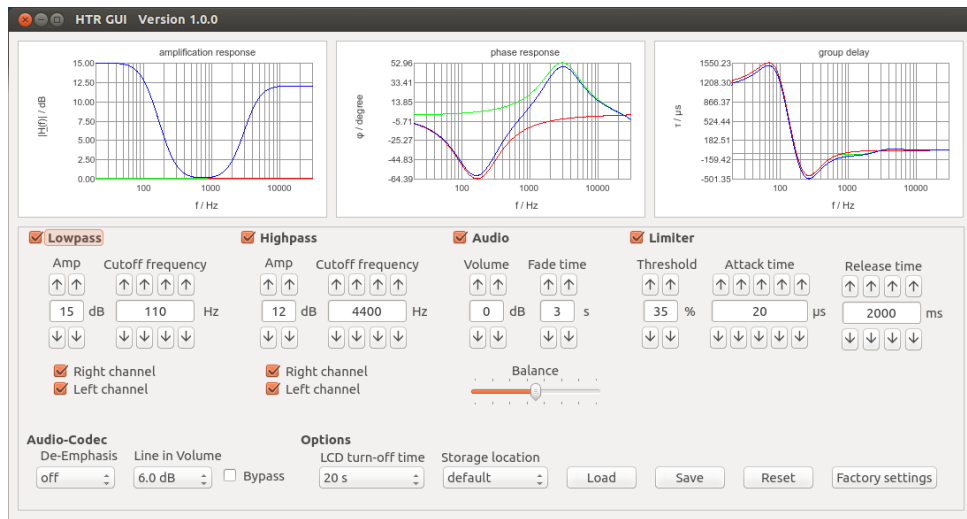


Bild 18: TCGUIP - Dynamic processor window



(a) Version 1.0



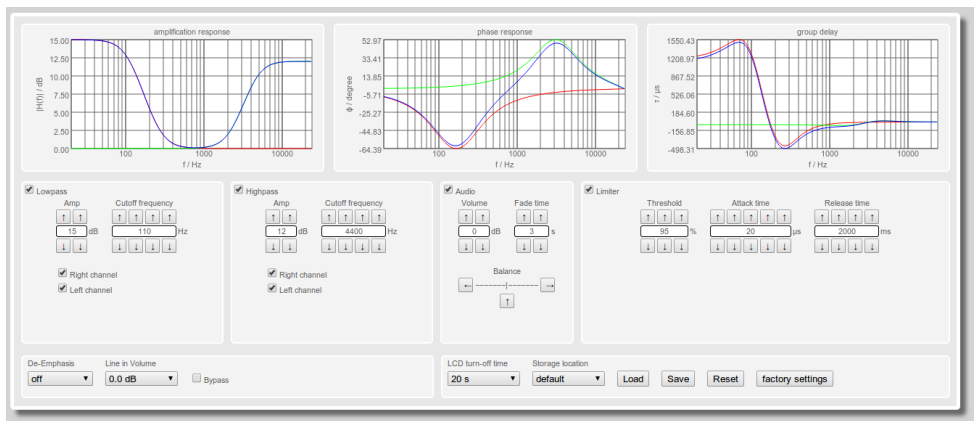
(b) Version 1.3

Bild 19: TCGUIP - Main window

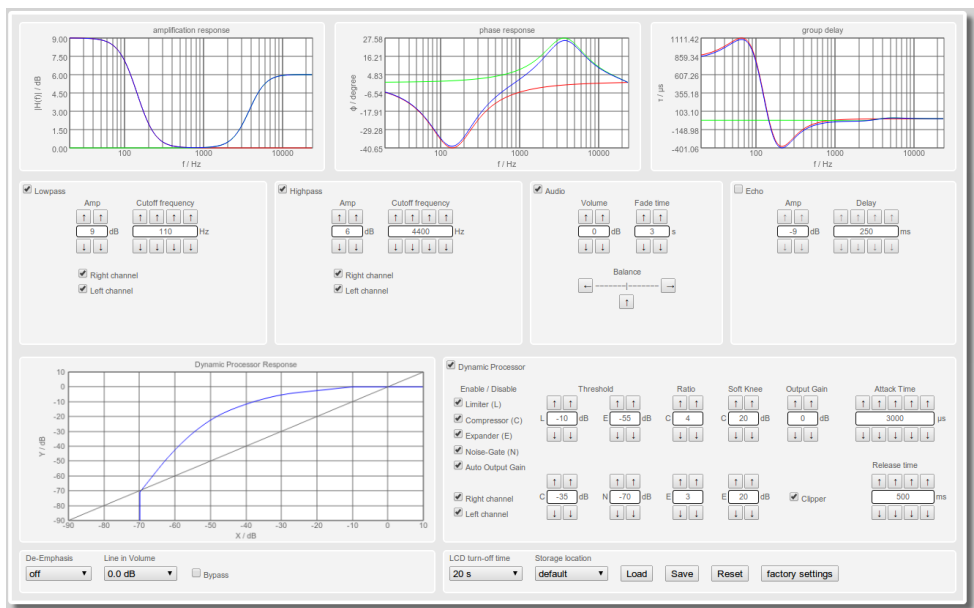
Für das Dynamic processor window wurden zwei Klassen dem TCGUIP hinzugefügt. Die Klasse *DynamicProcessorWindow* stellt die Methoden und Attribute zur Verfügung, um das Dynamic processor window anzuzeigen und die Anzeigen zu aktualisieren und auf Eingaben des Anwenders zu reagieren. Die Klasse *DynamicProcessorGraph* stellt Methoden und Attribute bereit, um die Kennlinie des Dynamic Processors entsprechend dessen Zustandes anzuzeigen.

4.4. Das Tone Control Webinterface Program (TCWIP)

Beim TCWIP sieht es ähnlich aus, wie beim TCGUIP. Auch hier wurde die Benutzeroberfläche und die Kommunikation mit der CU angepasst. Es wurde das Panel des Limiters entfernt und durch das Panel des Echos ersetzt. Unterhalb der Panels des Tiefen- und Höhen-Shelving-Filters, der Lautstärkeeinstellung und des Echos wurde eine weitere Zeile für den Dynamic Processor eingebracht. Auf der rechten Seite finden sich die Einstellmöglichkeiten und auf der linken Seite wird die Kennlinie des Dynamic Processors entsprechend dessen Zustandes angezeigt. Darunter sind wie gewohnt die Panels des Audio-Codexs und der Optionen zu finden. Bild 20 zeigt im Vergleich die Oberfläche der Version 1.0 und die der Version 1.3.



(a) Version 1.0



(b) Version 1.3

Bild 20: TCWIP - Benutzeroberfläche

5. Die Messergebnisse

Zum Zeitpunkt der Niederschrift dieses Dokumentes wurden noch keine Messungen vorgenommen. Es wurde lediglich auditiv geprüft.

6. Tests und erkannte Fehler

Tests wurden in der Form vorgenommen, dass die geänderten und neu hinzugekommenen Parameter in der Form überprüft wurden, dass die Änderungen der Parameter an den Benutzerschnittstellen durch den Anwender zu Parameteränderungen in der Audiosignalverarbeitung führen. Bei diesen Tests konnte kein Fehler festgestellt werden.

Weiterhin wurde getestet, dass die Konfiguration mit den neuen und geänderten Parametern vollständig abgespeichert und geladen wird. Auch hier ließen sich keine Fehler feststellen.

7. Zusammenfassung und Ausblick

Mit dem Hinzufügen des Clippers und des Echo-Effektes und dem Austausch des Limiters durch den Dynamic Processor konnte gezeigt werden, dass es in einfacher Weise möglich ist die Klangregelung zu erweitern. Dazu wurden in der Audiosignalverarbeitung entsprechende Blöcke erstellt und dieser hinzugefügt und die Benutzerschnittstellen angepasst.

Weitere Verbesserungen können sein, den Quellcode auf weitere Optimierungen zu untersuchen, die mathematischen Funktionen der math-Bibliothek durch andere schnellere Algorithmen zu tauschen. Hier wäre eine Untersuchung der Logarithmusfunktion interessant. Weiter sollte die Effektivität des Compilers untersucht werden, um so zu ersehen, bei welchen Bestandteilen des Quellcodes es sich lohnen könnte diese durch Assembler-Anweisungen zu ersetzen.

Weitere Verbesserungen könnten in der Nutzung von weiteren Look-Up-Tables sein, um den Rechenaufwand weiter zu minimieren.

Literatur

- [1] ZÖLZER, Udo: *Digitale Audiosignalverarbeitung*. 3. überarbeitete und erweiterte Auflage 2005. Vieweg+Teubner GWV Fachverlage GmbH, Wiesbaden, 2008. – ISBN 978-3-519-26180-3
- [2] BRONSTEIN, I.N. ; SEMENDJAJEW, K.A. ; MUSIOL, G. ; MÜHLIG, H.: *Taschenbuch der Mathematik*. 7. Auflage. Wissenschaftlicher Verlag Harri Deutsch GmbH, 2008. – ISBN 978-3-8171-2017-8

Anhang

Anhangsverzeichnis

Anhangsverzeichnis	2
A. Steuersignale	3

A. Steuersignale

Im folgenden sind die Steuersignale mit ihren Eigenschaften aufgelistet, welche zum Einstellen der Parameter vom Mikroprozessor zum Signalprozessor, über den I²C-Bus übermittelt werden.

Bezeichnung: Eingangslautstärke (Line in Volume)
 Symbol: G_{ADC}
 Einstellbereich: -34,5 dB bis 12,0 dB in 1,5-dB-Schritten
 Steuerparameter: $s_{G_{ADC}}$
 Registeradresse: 0
 Wortbreite: 8 bit
 Steuerwerte: $s_{G_{ADC}} = \frac{G_{ADC} + 34,5}{1,5}$
 Beschreibung: Die Eingangslautstärke (Line in Volume) G_{ADC} ist die Lautstärkeeinstellung, welche im Audio-Codec für das Eingangssignal vorgegeben werden kann.

Bezeichnung: Deakzentuierung (De-Emphasis)
 Einstellungen: off, 32 kHz, 44,1 kHz, 48 kHz
 Steuerparameter: s_d
 Registeradresse: 1
 Wortbreite: 8 bit
 Steuerwerte: 0 - off
 1 - 32 kHz
 2 - 44,1 kHz
 3 - 48 kHz
 Beschreibung: Das De-Emphasis ist ein Deakzentuierungsfiter, welches eine Akzentuierung bei der Aufnahme oder Übertragung aufhebt.

Bezeichnung: DAC select
 Einstellungen: enable, disable
 Steuerparameter: $s_{DAC_{en}}$
 Registeradresse: 2
 Wortbreite: 8 bit
 Steuerwerte: 0 - disable
 1 - enable
 Beschreibung: Der Steuerparameter *DAC select* gibt an, ob das DAC zugeschaltet sein soll oder nicht.

Bezeichnung: Bypass
Einstellungen: enable, disable
Steuerparameter: s_b
Registeradresse: 3
Wortbreite: 8 bit
Steuerwerte: 0 - disable
1 - enable
Beschreibung: Der Bypass ist im Audio-Codec verankert und schleift das Eingangssignal direkt auf den Ausgang. Ist der DAC noch zugeschaltet, dann werden beide Signale auf den Ausgang gelegt und überlagern sich. Somit muss für das verwenden des Bypass nicht nur der Steuerparameter s_b geschaltet werden, sondern auch der Steuerparameter $s_{DAC_{en}}$.

Bezeichnung: Lowpass Bypass
Einstellungen: enable, disable
Steuerparameter: $s_{lp_{en}}$
Registeradresse: 4
Wortbreite: 8 bit
Steuerwerte: 0 - disable
1 - enable
Beschreibung: Über den Steuerparameter *Lowpass Bypass* kann der Tiefen-Shelving-Filter in den Audiosignalweg zugeschaltet werden oder aus diesem herausgenommen werden.

Bezeichnung: Lowpass left channel bypass
Einstellungen: enable, disable
Steuerparameter: $s_{lp_{l_{en}}}$
Registeradresse: 5
Wortbreite: 8 bit
Steuerwerte: 0 - disable
1 - enable
Beschreibung: Über den Steuerparameter *Lowpass left channel bypass* kann das Tiefen-Shelving-Filter in den Signalweg des linken Kanals des Stereosignal zugeschaltet werden oder aus diesem herausgenommen werden.

Bezeichnung: Lowpass right channel bypass
Einstellungen: enable, disable
Steuerparameter: $s_{lp_{r_{en}}}$
Registeradresse: 6
Wortbreite: 8 bit
Steuerwerte: 0 - disable
1 - enable
Beschreibung: Über den Steuerparameter *Lowpass right channel bypass* kann das Tiefen-Shelving-Filter in den Signalweg des rechten Kanals des Stereosignal zugeschaltet oder aus diesem herausgenommen werden.

Bezeichnung: Lowpass cutoff frequency
 Einstellbereich: 20 Hz bis 1000 Hz in 1-Hz-Schritten
 Symbol: $f_{c_{lp}}$
 Steuerparameter: $s_{f_{c_{lp}}}$
 Registeradresse: 7 (oberes Byte)
 8 (unteres Byte)
 Wortbreite: 16 bit
 Steuerwerte: $s_{f_{c_{lp}}} = f_{c_{lp}}$
 Beschreibung: Über den Steuerparameter $s_{f_{c_{lp}}}$ werden die Werte der Grenzfrequenz des Tiefen-Shelving-Filters übermittelt.

Bezeichnung: Lowpass amplification factor
 Einstellbereich: -30 dB bis 30 dB in 1-dB-Schritten
 Symbol: G_{lp}
 Steuerparameter: $s_{G_{lp}}$
 Registeradresse: 9
 Wortbreite: 8 bit
 Steuerwerte: $s_{G_{lp}} = 30 + G_{lp}$
 Beschreibung: Über den Steuerparameter $s_{G_{lp}}$ wird der logarithmische Verstärkungsfaktor für den Tiefen-Shelving-Filter übertragen.

Bezeichnung: Highpass Bypass
 Einstellungen: enable, disable
 Steuerparameter: $s_{hp_{en}}$
 Registeradresse: 10
 Wortbreite: 8 bit
 Steuerwerte: 0 - disable
 1 - enable
 Beschreibung: Über den Steuerparameter *Highpass Bypass* kann der Höhen-Shelving-Filter in den Audiosignalweg zugeschaltet werden oder aus diesem herausgenommen werden.

Bezeichnung: Highpass left channel bypass
 Einstellungen: enable, disable
 Steuerparameter: $s_{hp_{len}}$
 Registeradresse: 11
 Wortbreite: 8 bit
 Steuerwerte: 0 - disable
 1 - enable
 Beschreibung: Über den Steuerparameter *Highpass left channel bypass* kann das Höhen-Shelving-Filter in den Signalweg des linken Kanals des Stereosignals zugeschaltet oder aus diesem herausgenommen werden.

Bezeichnung: Highpass right channel bypass
 Einstellungen: enable, disable
 Steuerparameter: s_{hpren}
 Registeradresse: 12
 Wortbreite: 8 bit
 Steuerwerte: 0 - disable
 1 - enable
 Beschreibung: Über den Steuerparameter *Highpass right channel bypass* kann das Höhen-Shelving-Filter in den Signalweg des rechten Kanals des Stereosignal zugeschaltet oder aus diesem herausgenommen werden.

Bezeichnung: Highpass cutoff frequency
 Einstellbereich: 1000 Hz bis 16000 Hz in 1-Hz-Schritten
 Symbol: $f_{c_{hp}}$
 Steuerparameter: $s_{f_{c_{hp}}}$
 Registeradresse: 13 (oberes Byte)
 14 (unteres Byte)
 Wortbreite: 16 bit
 Steuerwerte: $s_{f_{c_{hp}}} = f_{c_{hp}}$
 Beschreibung: Über den Steuerparameter $s_{f_{c_{hp}}}$ werden die Werte der Grenzfrequenz des Höhen-Shelving-Filters übermittelt.

Bezeichnung: Highpass amplification factor
 Einstellbereich: -30 dB bis 30 dB in 1-dB-Schritten
 Symbol: G_{hp}
 Steuerparameter: $s_{G_{hp}}$
 Registeradresse: 15
 Wortbreite: 8 bit
 Steuerwerte: $s_{G_{hp}} = 30 + G_{hp}$
 Beschreibung: Über den Steuerparameter $s_{G_{hp}}$ wird der logarithmische Verstärkungsfaktor für den Höhen-Shelving-Filter übertragen.

Bezeichnung: Volume
 Einstellbereich: -30 dB bis 30 dB in 1-dB-Schritten
 Symbol: G_V
 Steuerparameter: s_{G_V}
 Registeradresse: 16
 Wortbreite: 8 bit
 Steuerwerte: $s_{G_V} = 30 + G_V$
 Beschreibung: Mit dem Steuerparameter Volume wird der logarithmische Verstärkungsfaktor G_V für den Block *Volume* der Lautstärkeinstellung übertragen.

Bezeichnung: Fade time
 Einstellbereich: 0 s bis 10 s in 1-s-Schritten
 Symbol: t_F
 Steuerparameter: s_{t_F}
 Registeradresse: 17
 Wortbreite: 8 bit
 Steuerwerte: $s_{t_F} = t_F$
 Beschreibung: Die Ein- und Ausblendzeit t_F gibt an, wie lange das Ein- und Ausblenden beim Zu- oder Abschalten der Stummschaltung (Mute) dauern soll.

Bezeichnung: Balance
 Einstellbereich: -7 bis 7 in 1er-Schritten
 Symbol: B
 Steuerparameter: s_B
 Registeradresse: 18
 Wortbreite: 8 bit
 Steuerwerte: $s_B = B$
 Beschreibung: Der Balancewert der im Bereich von -7 bis 7 als ganze natürliche Zahl angegeben wird, repräsentiert das Lautstärkeverhältnis zwischen rechtem und linken Kanal des Stereosignals.

Bezeichnung: Mute
 Einstellungen: enable, disable
 Symbol: c_m
 Steuerparameter: s_{c_m}
 Registeradresse: 19
 Wortbreite: 8 bit
 Steuerwerte: $s_{c_m} = c_m$
 Beschreibung: Der Wert c_m gibt mit 0 an, dass die Stummschaltung abgeschaltet und mit 1, dass diese zugeschaltet ist. Dieser Wert wird direkt über den Steuerparameter s_{c_m} übertragen.

Bezeichnung: Dynamic Processor Bypass
 Einstellungen: enable, disable
 Symbol: D_{en}
 Steuerparameter: $s_{D_{en}}$
 Registeradresse: 20
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{en}} = D_{en}$
 Beschreibung: Mit dem Steuerparameter $s_{D_{en}}$ wird an den Signalprozessor übertragen, ob der Dynamic Processor in den Signalweg zugeschaltet oder aus diesem herausgenommen werden soll.

Bezeichnung: Dynamic Processor Left Channel Bypass
Einstellbereich: enable, disable
Symbol: D_{Len}
Steuerparameter: $s_{D_{Len}}$
Registeradresse: 21
Wortbreite: 8 bit
Steuerwerte: $s_{D_{Len}} = D_{Len}$
Beschreibung: Mit diesem Signal kann angegeben werden, ob der linke Kanal den Dynamic Processor durchlaufen soll oder nicht.

Bezeichnung: Dynamic Processor Right Channel Bypass
Einstellbereich: enable, disable
Symbol: D_{Ren}
Steuerparameter: $s_{D_{Ren}}$
Registeradresse: 22
Wortbreite: 8 bit
Steuerwerte: $s_{D_{Ren}} = D_{Ren}$
Beschreibung: Mit diesem Signal kann angegeben werden, ob der rechte Kanal den Dynamic Processor durchlaufen soll oder nicht.

Bezeichnung: Dynamic Processor Limiter Bypass
Einstellbereich: enable, disable
Symbol: D_{Len}
Steuerparameter: $s_{D_{Len}}$
Registeradresse: 23
Wortbreite: 8 bit
Steuerwerte: $s_{D_{Len}} = D_{Len}$
Beschreibung: Mit diesem Signal kann angegeben werden, ob der Limiter des Dynamic Processors aktiviert oder deaktiviert sein soll.

Bezeichnung: Dynamic Processor Compressor Bypass
Einstellbereich: enable, disable
Symbol: D_{Cen}
Steuerparameter: $s_{D_{Cen}}$
Registeradresse: 24
Wortbreite: 8 bit
Steuerwerte: $s_{D_{Cen}} = D_{Cen}$
Beschreibung: Mit diesem Signal kann angegeben werden, ob der Compressor des Dynamic Processors aktiviert oder deaktiviert sein soll.

Bezeichnung: Dynamic Processor Expander Bypass
 Einstellbereich: enable, disable
 Symbol: D_{En}
 Steuerparameter: $s_{D_{En}}$
 Registeradresse: 25
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{En}} = D_{En}$
 Beschreibung: Mit diesem Signal kann angegeben werden, ob der Expander des Dynamic Processors aktiviert oder deaktiviert sein soll.

Bezeichnung: Dynamic Processor Noise-Gate Bypass
 Einstellbereich: enable, disable
 Symbol: D_{Nen}
 Steuerparameter: $s_{D_{Nen}}$
 Registeradresse: 26
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{Nen}} = D_{Nen}$
 Beschreibung: Mit diesem Signal kann angegeben werden, ob das Noise-Gate des Dynamic Processors aktiviert oder deaktiviert sein soll.

Bezeichnung: Dynamic Processor Auto Output Gain
 Einstellbereich: enable, disable
 Symbol: D_{Gen}
 Steuerparameter: $s_{D_{Gen}}$
 Registeradresse: 27
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{Gen}} = D_{Gen}$
 Beschreibung: Mit diesem Signal kann angegeben werden, ob das Auto Output Gain des Dynamic Processors aktiviert oder deaktiviert sein soll.

Bezeichnung: Dynamic Processor Limiter Threshold
 Einstellbereich: -90 dB bis 10 dB in 1-dB-Schritten
 Symbol: D_{LT}
 Steuerparameter: $s_{D_{LT}}$
 Registeradresse: 28
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{LT}} = D_{LT}$
 Beschreibung: Der Limiter beginnt bei einem Eingangspegel oberhalb der eingestellten Limiterschwelle L_T mit der Steigung $L_S = 1$ zu begrenzen. Das Signal $s_{D_{LT}}$ gibt diese Schwelle in dB an.

Bezeichnung: Dynamic Processor Compressor Threshold
 Einstellbereich: -90 dB bis 10 dB in 1-dB-Schritten
 Symbol: D_{CT}
 Steuerparameter: $s_{D_{CT}}$
 Registeradresse: 29
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{CT}} = D_{CT}$
 Beschreibung: Der Compressor beginnt bei einem Eingangsspiegel oberhalb der eingestellten Compressor-Schwelle C_T bis zur Limiter-Schwelle L_T mit der Steigung C_S zu komprimieren. Das Signal $s_{D_{CT}}$ gibt diese Schwelle in dB an.

Bezeichnung: Dynamic Processor Expander Threshold
 Einstellbereich: -90 dB bis 10 dB in 1-dB-Schritten
 Symbol: D_{ET}
 Steuerparameter: $s_{D_{ET}}$
 Registeradresse: 30
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{ET}} = D_{ET}$
 Beschreibung: Der Expander beginnt bei einem Eingangsspiegel unterhalb der eingestellten Expander-Schwelle E_T bis zur Noise-Gate-Schwelle N_T mit der Steigung E_S zu expandieren. Das Signal $s_{D_{ET}}$ gibt diese Schwelle in dB an.

Bezeichnung: Dynamic Processor Noise-Gate Threshold
 Einstellbereich: -90 dB bis 10 dB in 1-dB-Schritten
 Symbol: D_{NT}
 Steuerparameter: $s_{D_{NT}}$
 Registeradresse: 31
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{NT}} = D_{NT}$
 Beschreibung: Das Noise-Gate beginnt bei einem Eingangsspiegel unterhalb der eingestellten Noise-Gate-Schwelle N_T mit der Steigung $N_S = -10^{12}$ zu arbeiten. Das Signal $s_{D_{NT}}$ gibt diese Schwelle in dB an.

Bezeichnung: Dynamic Processor Compressor Ratio
 Einstellbereich: 1:1 bis 40:1 in 0,1-Schritten
 Symbol: D_{CR}
 Steuerparameter: $s_{D_{CR}}$
 Registeradresse: 32
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{CR}} = 10 \cdot D_{CR}$
 Beschreibung: Die Compressor Ratio gibt an wie stark das Audiosignal komprimiert werden soll. Die Angabe erfolgt in der Form $D_{CR}:1$. Dies bedeutet, dass bei einer eingestellten Ratio $D_{CR} = 4$ eine Eingangspiegeländerung von beispielsweise 4 dB zu einer Ausgangspiegeländerung von 1 dB führt.

Bezeichnung: Dynamic Processor Expander Ratio
 Einstellbereich: 1:1 bis 1:40 in 0,1-Schritten
 Symbol: D_{ER}
 Steuerparameter: $s_{D_{ER}}$
 Registeradresse: 33
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{ER}} = 10 \cdot D_{ER}$
 Beschreibung: Die Expander Ratio gibt an wie stark das Audiosignal expandiert werden soll. Die Angabe erfolgt in der Form 1: D_{ER} . Dies bedeutet, dass bei einer eingestellten Ratio $D_{ER} = 3$ eine Eingangspegeländerung von beispielsweise 1 dB zu einer Ausgangspegeländerung von 3 dB führt.

Bezeichnung: Dynamic Processor Compressor Soft-Knee wideness
 Einstellbereich: 0 dB bis 40 dB in 1-dB-Schritten
 Symbol: D_{CK}
 Steuerparameter: $s_{D_{CK}}$
 Registeradresse: 34
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{CK}} = D_{CK}$
 Beschreibung: Mit der Soft-Knee-Breite wird in dB angegeben in welchem Bereich das Soft-Knee arbeiten soll. Dabei wird die Compressor-Schwelle C_T als Mittelpunkt verwendet. Der Startpunkt liegt bei einem Eingangspegel von $C_T - \frac{C_K}{2}$ und der Endpunkt bei einem Eingangspegel von $C_T + \frac{C_K}{2}$. Das Soft-Knee ist so aufgebaut, dass mit einer Ratio von 1:1 begonnen wird und am Endpunkt die Ratio C_R erreicht wird.

Bezeichnung: Dynamic Processor Expander Soft-Knee wideness
 Einstellbereich: 0 dB bis 40 dB in 1-dB-Schritten
 Symbol: D_{EK}
 Steuerparameter: $s_{D_{EK}}$
 Registeradresse: 35
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{EK}} = D_{EK}$
 Beschreibung: Mit der Soft-Knee-Breite wird in dB angegeben in welchem Bereich das Soft-Knee arbeiten soll. Dabei wird die Expander-Schwelle C_T als Mittelpunkt verwendet. Der Startpunkt liegt bei einem Eingangspegel von $E_T + \frac{E_K}{2}$ und der Endpunkt bei einem Eingangspegel von $E_T - \frac{E_K}{2}$. Das Soft-Knee ist so aufgebaut, dass mit einer Ratio von 1:1 begonnen wird und am Endpunkt die Ratio E_R erreicht wird.

Bezeichnung: Dynamic Processor Manuell Output Gain
 Einstellbereich: -90 dB bis 90 dB in 1-dB-Schritten
 Symbol: D_{Gm}
 Steuerparameter: $s_{D_{Gm}}$
 Registeradresse: 36
 Wortbreite: 8 bit
 Steuerwerte: $s_{D_{Gm}} = D_{Gm}$
 Beschreibung: Bei deaktiviertem Auto Output Gain ist dies der logarithmische Verstärkungsfaktor der dem Verstärkungsfaktor der sich aus der Dynamic Processor Kennlinie ergibt hinzugefügt wird, um so einen Hub in der Kennlinie zu erreichen.

Bezeichnung: Dynamic Processor Attack time
 Einstellbereich: 30 μ s bis 500 ms in 10- μ s-Schritten
 Symbol: D_{ta}
 Steuerparameter: $s_{D_{ta}}$
 Registeradresse: 37 (oberes Byte)
 38 (unteres Byte)
 Wortbreite: 16 bit
 Steuerwerte: $s_{D_{ta}} = \frac{D_{ta}}{10}$
 Beschreibung: Ansprechzeit t_a der Komponenten des Dynamic Processors mit Ausnahme des Limiters. Dieser hat eine fest vorgegebene Ansprechzeit. D_{ta} wird in Mikrosekunden angegeben.

Bezeichnung: Dynamic Processor Release time
 Einstellbereich: 1 ms bis 5 s in 1-ms-Schritten
 Symbol: D_{tr}
 Steuerparameter: $s_{D_{tr}}$
 Registeradresse: 39 (oberes Byte)
 40 (unteres Byte)
 Wortbreite: 16 bit
 Steuerwerte: $s_{D_{tr}} = D_{tr}$
 Beschreibung: Rücklaufzeit t_r der Komponenten des Dynamic Processors mit Ausnahme des Limiters. Dieser hat eine fest vorgegebene Rücklaufzeit. D_{tr} wird in Millisekunden angegeben.

Bezeichnung: Echo Bypass
 Einstellbereich: enable, disable
 Symbol: E_{en}
 Steuerparameter: $s_{E_{en}}$
 Registeradresse: 41
 Wortbreite: 8 bit
 Steuerwerte: $s_{E_{en}} = D_{en}$
 Beschreibung: Gibt an ob das Audiosignal den Echo-Effekt durchlaufen soll oder nicht.

Bezeichnung: Echo amplification factor
 Einstellbereich: -20 dB bis -1 dB in 1-dB-Schritten
 Symbol: E_{GE}
 Steuerparameter: $s_{E_{GE}}$
 Registeradresse: 42
 Wortbreite: 8 bit
 Steuerwerte: $s_{E_{GE}} = D_{GE}$
 Beschreibung: Logarithmischer Verstärkungsfaktor des Echos in dB.

Bezeichnung: Echo Delay
 Einstellbereich: 1 ms bis 1000 ms in 1-ms-Schritten
 Symbol: E_{tE}
 Steuerparameter: $s_{E_{tE}}$
 Registeradresse: 43 (oberes Byte)
 44 (unteres Byte)
 Wortbreite: 16 bit
 Steuerwerte: $s_{E_{tE}} = E_{tE}$
 Beschreibung: Verzögerungszeit t_E des Echo-Effektes in Millisekunden.

Bezeichnung: Clipper Bypass
 Einstellbereich: enable, disable
 Symbol: $clipper_{en}$
 Steuerparameter: $s_{clipper_{en}}$
 Registeradresse: 45
 Wortbreite: 8 bit
 Steuerwerte: $s_{clipper_{en}} = clipper_{en}$
 Beschreibung: Gibt an ob das Audiosignal den Clipper durchlaufen soll oder nicht.

Die folgenden Steuerparameter werden nicht an den Signalprozessor übermittelt, sondern von den Benutzerschnittstellen an den Mikrocontroller übertragen, um diesen in seinem Verhalten zu steuern.

Bezeichnung: LCD turn-off time
Einstellbereich: 0 s bis 10 s in 1-Sekunden-Schritten
Symbol: t_{LCD}
Steuerparameter: s_{LCD}
Wortbreite: 8 bit
Steuerwerte: $s_{LCD} = t_{LCD}$
Beschreibung: Die *LCD turn-off time* t_{LCD} gibt an, wie lange die Hintergrundbeleuchtung des LCD leuchten soll, wenn keine weitere Eingaben vom Benutzer gemacht werden. Nach Ablauf dieser Zeit, wird die Hintergrundbeleuchtung abgeschaltet.

Bezeichnung: Storage location
Einstellbereich: 0 bis 9
Symbol: $storage$
Steuerparameter: $s_{storage}$
Wortbreite: 8 bit
Steuerwerte: $s_{storage} = storage$
Beschreibung: Die Angabe des Speicherortes für das Speichern der Konfiguration erfolgt über das Steuersignal $s_{storage}$.

Bezeichnung: Load
Einstellbereich: beliebig
Symbol: $load$
Steuerparameter: s_{load}
Wortbreite: 8 bit
Steuerwerte: $s_{load} = load$
Beschreibung: Wird der Steuerparameter s_{load} mit einem beliebigen Wert von einer Benutzerschnittstelle an den Mikrocontroller gesendet, so veranlasst dieser das Laden der am eingestellten Speicherort abgelegten Konfiguration.

Bezeichnung: Save
Einstellbereich: beliebig
Symbol: $save$
Steuerparameter: s_{save}
Wortbreite: 8 bit
Steuerwerte: $s_{save} = save$
Beschreibung: Wird der Steuerparameter s_{save} mit einem beliebigen Wert von einer Benutzerschnittstelle an den Mikrocontroller gesendet, so veranlasst dieser das Speichern der aktuellen Konfiguration im angegebenen Speicherort.

Bezeichnung: Reset
Einstellbereich: beliebig
Symbol: *reset*
Steuerparameter: s_{reset}
Wortbreite: 8 bit
Steuerwerte: $s_{reset} = reset$
Beschreibung: Wird der Steuerparameter s_{reset} mit einem beliebigen Wert von einer Benutzerschnittstelle an den Mikrocontroller gesendet, so veranlasst dieser das Zurücksetzen des Signalprozessors und des Mikrocontrollers.

Bezeichnung: Factory settings
Einstellbereich: beliebig
Symbol: *factory*
Steuerparameter: $s_{factory}$
Wortbreite: 8 bit
Steuerwerte: $s_{factory} = factory$
Beschreibung: Wird der Steuerparameter $s_{factory}$ mit einem beliebigen Wert von einer Benutzerschnittstelle an den Mikrocontroller gesendet, so veranlasst dieser die Speicherung der Standardwerte auf den Speicherort 0 und anschließend das Zurücksetzen des Signalprozessors und des Mikrocontrollers.

