

Laborversuch 3

(freiwilliges Eigenstudium)

Kompandierung

Der Laborversuch "Kompandierung" hat das bei der Analog-/Digitalwandlung entstehende Quantisierungsrauschen und die damit verbundene Absenkung des Signal-/Rauschabstands eines Signals zum Gegenstand.

Unter Kompandierung versteht man ein Verfahren, bei dem über den gesamten Aussteuerungsbereich eines A/D-Wandlers durch eine besondere (nichtlineare) Vorverzerrung des Eingangssignals ein konstantes Signal-zu-Quantisierungsrauschleistungsverhältnis erzielt wird.

Kompandierung wird in vielen digitalen Übertragungs- und Signalverarbeitungssystemen eingesetzt, z.B. in der Telefontechnik.

1. Einführende Informationen

1.1 Kompression in der Telefontechnik

In digitalen Übertragungssystemen, wie z.B. der Telefontechnik, wird Kompandierung gleichzeitig dazu eingesetzt, die von digitalen Audiosignalen benötigte Datenrate durch eine nicht-lineare Quantisierung zu reduzieren. Für "Telefonsprachqualität" wird bei linearer Quantisierung eine Auflösung von mindestens 12 Bit benötigt. Durch entsprechende Kompandierung kann dieser Wert auf 8 Bit reduziert werden. Die erforderliche Kompression kann dabei prinzipiell an drei verschiedenen Stellen der Signalverarbeitungskette erfolgen:

1. durch analoge Kompression vor einem linearen 8-Bit-A/D-Wandler
2. durch einen 8-Bit-A/D-Wandler mit nichtlinearer Quantisierung
3. durch digitale Kompression nach einem linearen 12-Bit-A/D-Wandler

Im Versuch soll eine sendeseitige, digitale Kompression und eine empfangsseitige Expansion auf dem TMS320C6711 programmiert und mit Hilfe der Soundkarte im Labor-PC und entsprechender Windows-Audio-Software getestet werden. Um den Gewinn der Kompandierung besser beurteilen zu können, wird anstelle des Codecs auf dem DSK6711 (TLC320AD535) ein qualitativ höherwertiger Codec auf einer auf das DSK6711 aufgesteckten Audio-Daughter-Card eingesetzt. Die Beurteilung des Gewinns erfolgt nach Gehör mit qualitativ hochwertigen Wave-Dateien, kann aber auch mit Hilfe des Audio Analyzers UPD von Rohde & Schwarz messtechnisch erfasst werden.

1.2 Quell-Code-Vorgaben

Angelehnt an den letzten Laborversuch stehen Ihnen folgende Quellcode-Dateien im Projektordner 4 zur Verfügung:

`Project4\compand.c`

Rumpfprogramm, das von Ihnen erweitert werden soll

`Project4\init_mcbasp_codec.c`

Initialisierung des McBSP1 für die Audio-Daughter-Card

`Project4\vectors3.asm`

Interrupt-Vektor-Tabelle

`Project4\c6711dsk.h`

Include-Datei (C6711 und DSK Definitionen)

`Project4\dsk6711.cmd`

Linker-Konfigurationsdatei

1.3 Die Windows-Audio-Software

Auf den Labor-PCs steht Ihnen neben dem Windows Media Player zum Abspielen von CDs und Wave-Dateien auch wieder das Programmpaket AudioTester zur Verfügung (siehe Laborversuch 2).

2. Theorie der Kompandierung

2.1 Lineare Quantisierung

Bei der Analog-/Digitalwandlung kommt es zu einer Zeit- und Wertdiskretisierung des analogen Signals. Unter linearer Quantisierung versteht man die Wertdiskretisierung in Quantisierungsintervalle mit einer festen Stufenhöhe Q .

In der Regel verwenden A/D-Wandler eine symmetrische Quantisierungskennlinie, wie in Abbildung 2.1a) für das Beispiel einer A/D-Wortbreite von $w = 2$ Bit ($2^w = 4$ Stufen) dargestellt. Die analoge Eingangsspannung x wird im Beispiel einem von 4 Digitalworten $[x]_Q$ zugeordnet. Bei einer angelegten analogen Spannung von 0 kann es vorkommen, dass der A/D-Wandler aufgrund seines Eigenrauschens zwischen zwei Digitalworten hin und her springt, es entsteht ein sogenanntes granulares Rauschen

$$x = 0 \Rightarrow [x]_Q = \begin{cases} Q/2 & \text{mit Wahrscheinlichkeit } 1/2 \\ -Q/2 & \text{mit Wahrsch. } 1/2 \end{cases} \quad (2.1)$$

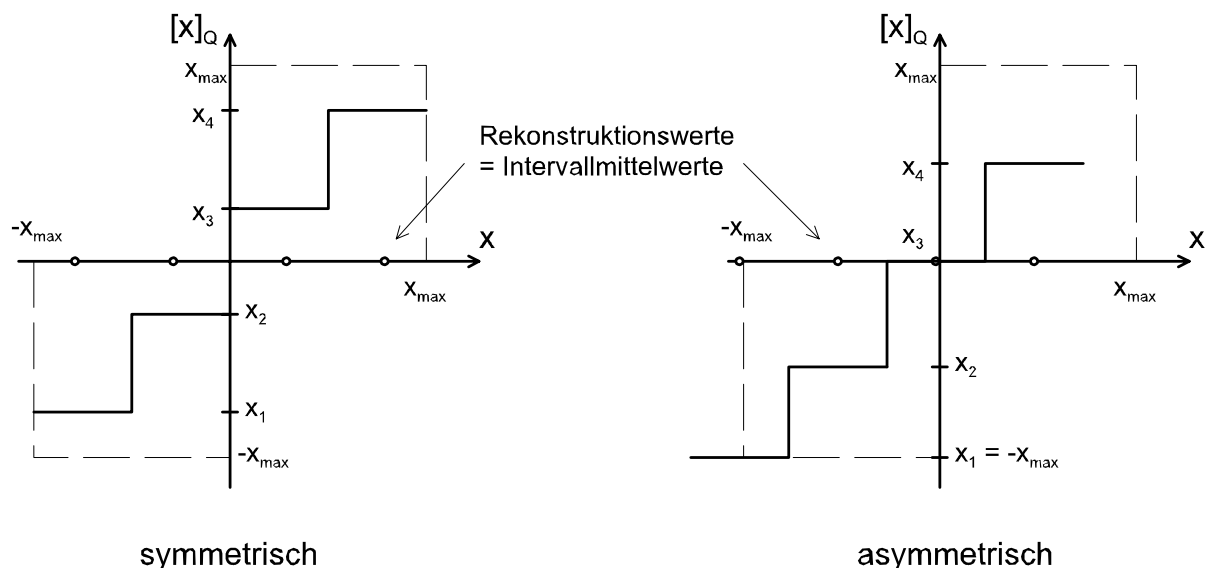


Abbildung 2.1: Symmetrische (a) und asymmetrische (b) Quantisierungskennlinien

SS 2008

Fakultät für Technik, Studiengänge EIT/TI

Prof. Dr.-Ing. Thorsten Benkner, Dipl.-Ing.(FH) Felix Becker

mit der Quantisierungsstufenhöhe $Q = 2 \cdot x_{\max} / 2^w$ und der Aussteuerungsgrenze x_{\max} des Wandlers.

Prinzipiell kann dieser Effekt durch eine asymmetrische Quantisierungskennlinie nach Abbildung 2.1b) vermieden werden, allerdings um den Nachteil eines asymmetrischen Aussteuerbereichs.

Durch den Quantisierungsvorgang kommt es zu einem Quantisierungsfehler, d.h., es ist keine exakte Rekonstruktion des Originalsignals aus seinen diskreten Amplitudenwerten mehr möglich. Der entstehende Fehler ε ist durch Gl.(2.2) als Differenz zwischen quantisiertem Signal $[x]_Q$ und Originalsignal x gegeben:

$$\varepsilon = [x]_Q - x \quad (2.2)$$

Abbildung 2.2 zeigt den Verlauf des Fehlersignals ε über der analogen Eingangsspannung x . Es ergibt sich ein sägezahnförmiger Verlauf mit dem Maximalwert

$$|\varepsilon| \leq \frac{Q}{2} = \frac{x_{\max}}{2^w} \quad (2.3)$$

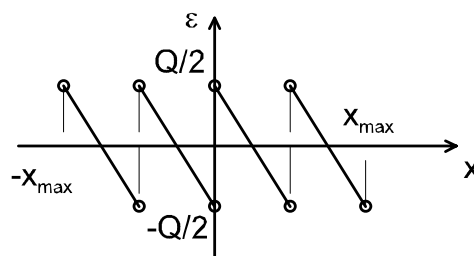


Abbildung 2.2: Verlauf der Fehlerspannung ε bei symmetrischer Quantisierungskennlinie

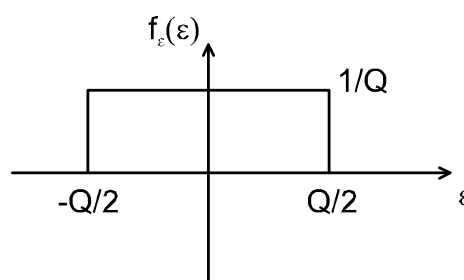


Abbildung 2.3: Verteilungsdichtefunktion des Fehlersignals ε

Der (mittelwertfreie) Quantisierungsfehler führt bei der Wiedergabe zu einem rauschähnlichen Signal, welches sich im Falle eines Audiosignals auch wie "normales" (Gaußsches) Rauschen anhört. Unter der Voraussetzung, dass ein (als rein zufällig anzusehendes) analoges Eingangssignal $x(t)$ innerhalb der einzelnen Quantisierungsstufen $[-Q/2, Q/2]$ gleichverteilt ist (siehe Abbildung 2.3), kann die Quantisierungsrauschleistung $E\{\varepsilon^2\}$ wie folgt berechnet werden:

SS 2008

Fakultät für Technik, Studiengänge EIT/TI

Prof. Dr.-Ing. Thorsten Benkner, Dipl.-Ing.(FH) Felix Becker

$$E\{\varepsilon^2\} = \int_{-\infty}^{+\infty} \varepsilon^2 \cdot f_{\varepsilon}(\varepsilon) d\varepsilon = \int_{-Q/2}^{Q/2} \varepsilon^2 \cdot \frac{1}{Q} d\varepsilon = \frac{\varepsilon^3}{3Q} \Big|_{-Q/2}^{Q/2} = \frac{2Q^3/8}{3Q} \quad (2.4)$$

$$\Rightarrow \boxed{E\{\varepsilon^2\} = \frac{Q^2}{12}}$$

Für die Wiedergabequalität ist das Signal-/Quantisierungsrauschleistungsverhältnis von Bedeutung. Wird der A/D-Wandler mit einem Sinussignal voll angesteuert,

$$x(t) = x_{\max} \cdot \sin(2\pi f_0 t + \varphi) \quad , f_0 \neq 0 \quad (2.5)$$

erhält man mit der (Nutz-)Signalleistung

$$S = x_{\text{eff}}^2 = \overline{x^2(t)} = x_{\max}^2 / 2 \quad (2.6)$$

und der Quantisierungsrauschleistung

$$N = \frac{Q^2}{12} \quad , \quad \frac{Q}{2} = \frac{x_{\max}}{2^w}$$

$$N = \left(\frac{x_{\max}}{2^w} \right)^2 / 3 \quad (2.7)$$

das Signal-/Quantisierungsrauschleistungsverhältnis

$$\frac{S}{N} = \frac{3}{2} \cdot 2^{2w} \quad (w: \text{Wortbreite des A/D-Wandlers}) \quad (2.8)$$

bzw. in logarithmischer Form

$$\frac{S}{N} \text{ dB} = 10 \cdot \log \frac{3}{2} + 2w \cdot 10 \cdot \log 2 = 1,8 + 6w \quad (2.9)$$

Aus einer um 1 Bit erhöhten Wortbreite des A/D-Wandlers resultiert also eine Zunahme des Signal-/Quantisierungsrauschleistungsverhältnisses von 6 dB. In Tabelle 2.1 sind die bei verschiedenen Wortbreiten erzielbaren, maximalen S/Ns angegeben.

w/bit	8	12	16
S/N [dB]	49,8	73,8	97,8

Tabelle 2.1: Signal-/Quantisierungsrauschleistungsverhältnisse

2.2 Nichtlineare Quantisierung

Das Signal-/Quantisierungsrauschleistungsverhältnis hängt von der Aussteuerung des A/D-Wandlers ab. Das S/N nach Gl.(2.8) bzw. Gl.(2.9) ist das maximal erzielbare, d.h., bei Vollaussteuerung des Wandlers. Bei geringerer Aussteuerung und demnach geringerer Signalleistung S nimmt das S/N entsprechend ab, da N konstant ist.

Fordert man ein über den gesamten Aussteuerungsbereich des A/D-Wandlers konstantes S/N, muss eine nichtlineare Quantisierungskennlinie realisiert werden, bei der für leise Signale ($|x|$ klein) eine feinere Quantisierungsstufenbreite benutzt wird als für laute Signale ($|x|$ groß).

So kommt man zu einer nichtlinearen Quantisierungskennlinie $Q_{NL}(x)$.

In der Praxis geht man nach dem Prinzip der "Momentanwertkompondierung" vor, d.h., das Eingangssignal $x(t)$ wird in einem Kompressor so vorverzerrt, dass kleine Signale stärker verstärkt werden als große. Es folgt dann eine lineare Quantisierung und nach der digitalen Übertragung über den Kanal im Empfänger ein Expander (Dekompressor) zur Korrektur der sendeseitigen Vorverzerrungen. Abbildung 2.4 zeigt das Funktionsprinzip.

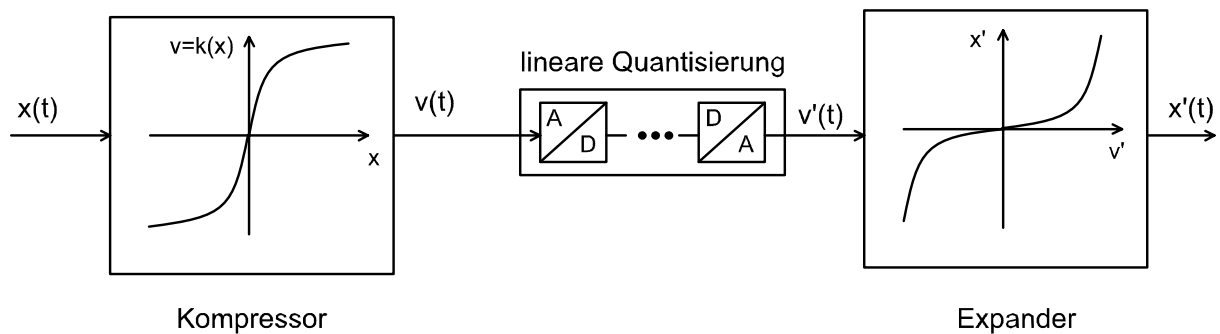


Abbildung 2.4: Kette aus Kompressor, A/D-Wandler, Kanal, D/A-Wandler und Expander

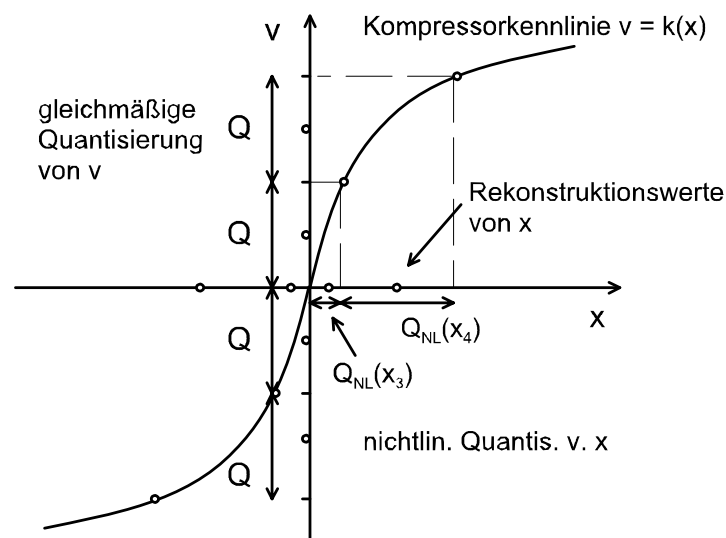


Abbildung 2.5: Entwurf der optimalen Kompressorkennlinie

SS 2008

Fakultät für Technik, Studiengänge EIT/TI

Prof. Dr.-Ing. Thorsten Benkner, Dipl.-Ing.(FH) Felix Becker

Im Kompressor entsteht eine nichtlineare Kennlinie dadurch, dass kleine Signalwerte "auseinandergezogen" und große Pegel "verdichtet" werden. Der Expander macht den Vorgang durch eine zum Kompressor inverse Kennlinie wieder rückgängig.

Es stellt sich nun die Frage nach der optimalen Kompressorkennlinie $v = k(x)$. Abbildung 2.5 zeigt das Entwurfsprinzip, bei dem kleine Eingangspegel x feiner quantisiert werden als große. Dies erfolgt dadurch, dass bei kleinen x die Quantisierungsstufenbreite $Q_{NL}(x)$ kleiner gewählt wird.

Die Intervallmittelwerte x_i der $Q_{NL}(x)$ werden auch wieder auf die Intervallmittelwerte der Ausgangsintervalle Q abgebildet. Wird die Kompressorkennlinie im Bereich eines Ein-/Ausgangsintervalls durch eine Gerade genähert, findet man den Zusammenhang

$$Q_{NL}(x_i) \approx Q \cdot \frac{1}{\left. \frac{dk(x)}{dx} \right|_{x=x_i}} = Q \cdot \left. \frac{dx}{dk(x)} \right|_{x=x_i} \quad (2.10)$$

mit

$$\frac{1}{\left. \frac{dk(x)}{dx} \right|_{x=x_i}} \equiv 1 / \text{Steigung}$$

Ziel ist es, das S/N näherungsweise unabhängig von der Aussteuerung zu machen, d.h., der maximale relative Quantisierungsfehler muss konstant sein:

$$\frac{Q_{NL}(x_i) / 2}{|x_i|} = |\alpha| = \text{const.} \quad (2.11)$$

Mit Gl.(2.10) in Gl.(2.11) erhält man

$$\frac{Q}{2} \cdot \frac{\left. \frac{dx}{dk(x)} \right|_{x=x_i}}{|x_i|} = \alpha \quad (2.12)$$

und damit

$$\left| \alpha \right| \cdot \left. \frac{dk(x)}{dx} \right|_{x=x_i} = \frac{Q}{2} \cdot \frac{1}{|x_i|} \quad \left(\int \frac{1}{x} dx = \ln|x| \right) \quad (2.13)$$

Gl.(2.13) ist eine Differentialgleichung für die gesuchte Funktion $k(x)$ mit der Lösung

$$k(x) = \begin{cases} x_{\max} + |c| \cdot \ln \frac{x}{x_{\max}} & \text{für } x > 0 \\ -x_{\max} - |c| \cdot \ln \frac{|x|}{x_{\max}} & \text{für } x < 0 \end{cases} \quad (c : \text{pos. Konstante, frei wählbar}) \quad (2.14)$$

Die Kennlinie des optimalen Kompressors ist also logarithmisch, unabhängig von der Verteilung von x . Sie kann sowohl für Sprach- als auch für Musiksingale eingesetzt werden. Bei der Realisierung stellt sich allerdings das Problem, dass die logarithmische Kennlinie im Expander nicht eindeutig umkehrbar ist (siehe Abbildung 2.6).

Abhilfe ist möglich, wenn der Bereich $x \in [-x_L, x_L]$ durch eine Gerade ersetzt wird, so dass sich ein Übergang ohne Knickstelle ergibt. Man gelangt so zur sog. A-Kennlinie ("A-Law"), die z.B. bei der digitalen Telefon-Sprachübertragung in Europa verwendet wird.

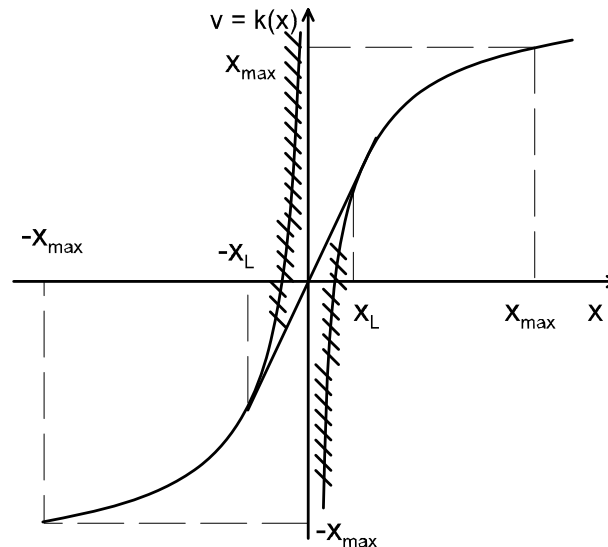


Abbildung 2.6: Modifikation der Kompressorkennlinie ("A-Law")

Die A-Kennlinie ("A-Law") hat den in Gl.(2.15) angegebenen Verlauf und ist eine ungerade Funktion, d.h., $k(-x) = -k(x)$.

$$k(x) = \begin{cases} \frac{A}{1 + \ln A} \cdot x & \text{für } 0 \leq x \leq x_L \\ x_{\max} \cdot \left[1 + \frac{\ln\left(\frac{x}{x_{\max}}\right)}{1 + \ln A} \right] & \text{für } x_L \leq x \leq x_{\max} \end{cases} \quad (2.15)$$

mit $A = \frac{x_{\max}}{x_L}$

Bei der Sprach-PCM im europäischen Fernsprechnet ist $A = 87,56$ (genormt).

Da die Kennliniensteigung bei kleinen x

$$\frac{A}{1 + \ln A} = 16$$

SS 2008

Fakultät für Technik, Studiengänge EIT/TI

Prof. Dr.-Ing. Thorsten Benkner, Dipl.-Ing.(FH) Felix Becker

beträgt, folgt daraus, dass kleine Signalanteile $|x| < x_L$ gegenüber einer linearen Quantisierung mit um 4 Bit ($2^4 = 16$) höherer Auflösung quantisiert werden.

Die Realisierung der A-Kennlinie erfolgt durch Annäherung über 13 Geradenabschnitte. Man erhält so die sog. "13-Segment-Kennlinie", die in Abbildung 2.7 gezeigt ist.

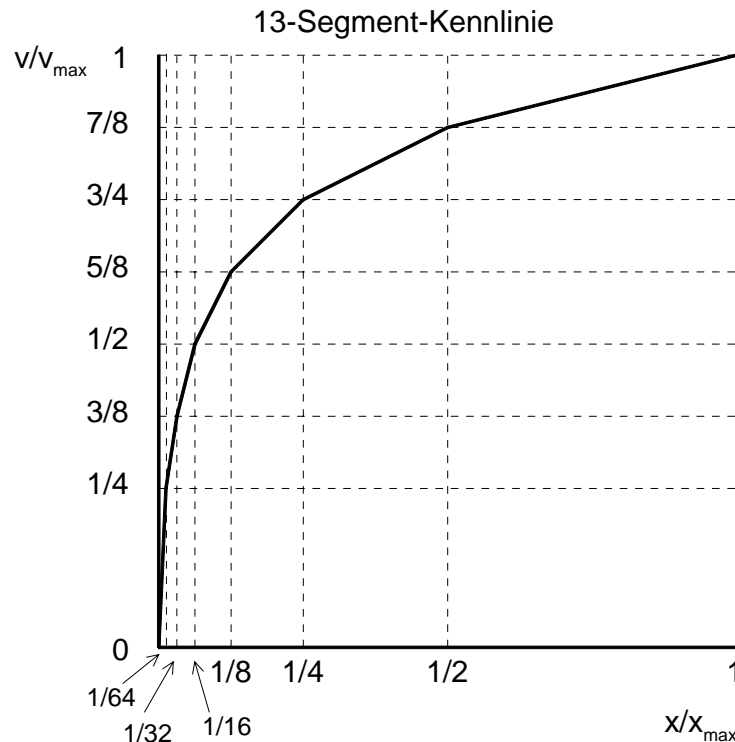


Abbildung 2.7: 13-Segment-Kennlinie

In USA und Japan wird bei der Sprachübertragung die μ -Law Kennlinie (15-Segment-Näherung) verwendet, bei der die beiden Äste der logarithmischen Kennlinie aufeinander zugeschoben werden, bis sie sich im Ursprung berühren.

Üblicherweise wird der Kompressor volldigital realisiert, indem die 12-Bit-Abtastwerte eines A/D-Wandlers nach Abbildung 2.8 in einem digitalen Kompressor in 8-Bit-Werte umgerechnet werden. Diese 12-Bit auf 8-Bit Umsetzung kann mit einer Lookup-Tabelle oder durch Bitschiebe- und Bitausblend-Operationen durchgeführt werden.

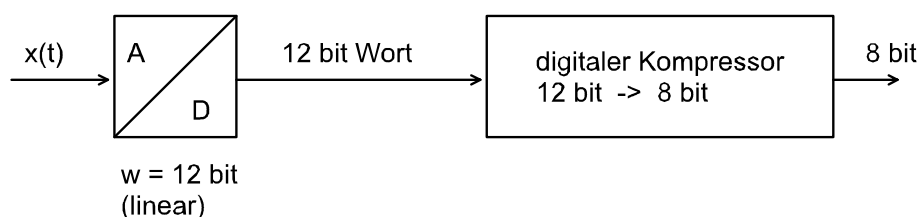


Abbildung 2.8: Digitaler Kompressor

Unser Laborversuch verwendet die Methode der Lookup-Tabelle, bei der keine zeitaufwendigen Rechenoperationen zur Laufzeit erforderlich sind.

Im Endergebnis erhält man mit dieser Technik eine um 4 Bit verbesserte Auflösung leiser Signale gegenüber einer linearen 8-Bit-Quantisierung, was einem S/N-Gewinn von 24 dB entspricht. Ferner ist das S/N nun über einen weiten Aussteuerungsbereich nahezu konstant.

3. Laborversuch

3.1 Digitale Kompression

Im Laborversuch soll zunächst das in Abbildung 2.8 dargestellte Verfahren realisiert werden. Hierzu ist eine Lookup-Tabelle zu programmieren, die $2^{12} = 4096$ Einträge besitzt.

(z.B.: `int comp[4096]` oder platzsparender: `unsigned char comp[4096]`)

Jeder Tabelleneintrag enthält ein 8-Bit-Wort, das den Verlauf der 13-Segment-Kompressor-kennlinie nachbildet. Abbildung 3.1 zeigt den Funktionsverlauf für 12-Bit Eingangsworte (= Tabellenindizes) und 8-Bit Ausgangsworte (= Tabelleneinträge).

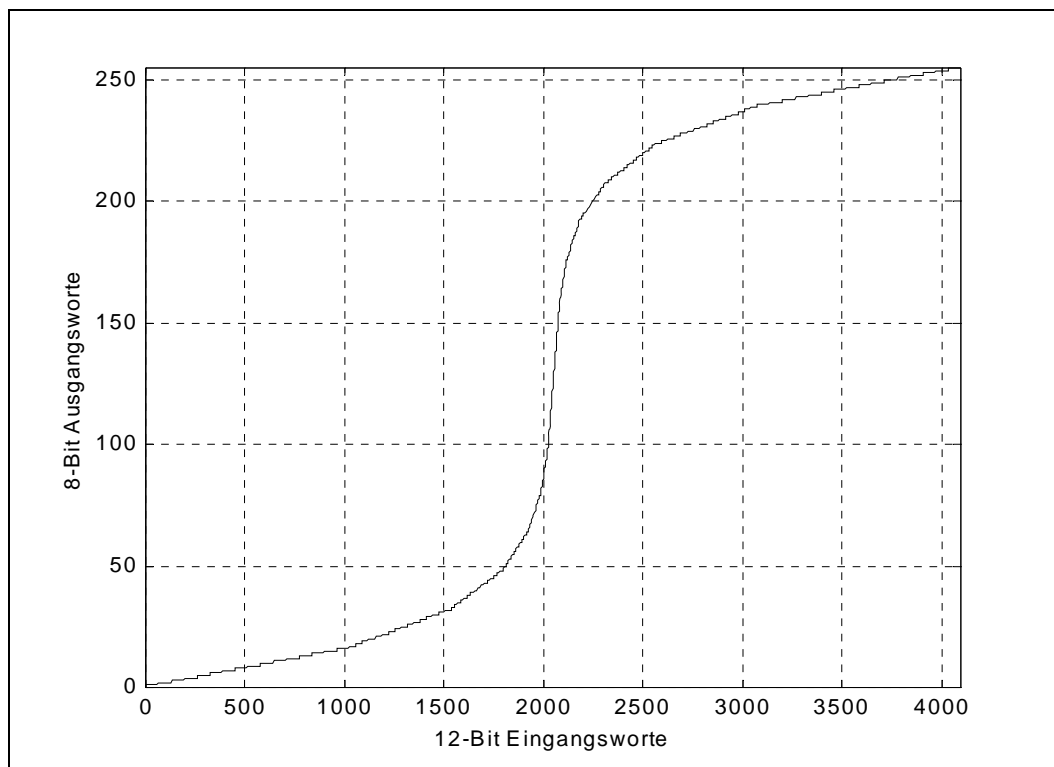


Abbildung 3.1: 13-Segment-Kennlinie der Lookup-Tabelle

Um den Gewinn der Kompondierung besser beurteilen zu können, wird – wie eingangs bereits erwähnt – anstelle des Codecs auf dem DSK6711 (TLC320AD535) ein qualitativ höherwertiger Codec auf einer auf das DSK6711 aufgesteckten Audio-Daughter-Card eingesetzt.

Dieser Codec (PCM3003) verarbeitet 20 Bit breite Abtastwerte und ist zweikanalig ausgeführt (Stereo L, R). Die Abtastrate beträgt 48 kHz. Der A/D-Wandler des Codecs liefert die Abtastwerte (abwechselnd L, R, L, R, L ...) in 2er-Komplement-Darstellung (K2), d.h., vorzeichenbehaftet.

Die 20 Bits werden durch die Receive-Interrupt-Routine vom Data-Receive-Register des McBSP1 in 32 Bit Integer-Werten linksbündig gelesen und durch arithmetisches Schieben aller Bits um 12 Bit nach rechts rechtsbündig und vorzeichenerweitert in einer globalen Variable abgelegt (siehe Code-Vorgaben in `Project4\compand.c`).

Im ersten Schritt der Kompression müssen diese Abtastwerte durch Addieren eines Offsets von 2^{19} (= 0x00080000) und anschließendem Verschieben aller Bits um 8 Bit nach rechts in vorzeichenlose 12-Bit-Worte konvertiert werden.

D.h., nach der Konvertierung entspricht ein Wert von $x = 2048$ dem Nullpunkt (0 Volt), $x = 0$ ist die negative Aussteuergrenze $-x_{\max}$, und $x = 4095$ ist die positive Aussteuergrenze x_{\max} .

Diese Werte sind jetzt die Eingangswerte (Indizes) für das Auslesen der komprimierten Werte aus der noch zu erstellenden Lookup-Tabelle (`comp[x]`). (Zweiter Schritt der Kompression.)

Die Lookup-Tabelle soll durch die Funktion `kennlinie_erstellen()` generiert werden.

Innerhalb dieser Funktion kann eine Schleife programmiert werden, die alle Werte von $k=0$ bis $k=4095$ durchläuft und so die zu generierenden Tabelleneinträge adressiert. Alternativ kann auch so vorgegangen werden, dass zunächst nur der positive Ast der 13-Segment-Kennlinie berechnet wird und anschließend der negative Ast durch eine entsprechende Spiegelung der Tabelleneinträge (ungerade Funktion) erstellt wird.

Die Kompressorkennlinie nach Gl.(2.15) wird in 13 Segmente unterteilt, die durch Geraden realisiert werden. Die Einteilung der x-Achse erfolgt durch Teilung um jeweils die Hälfte, wobei der Bereich $-1/64 \dots +1/64$ (entspricht den unteren 5 der 12 Bit) linear codiert wird.

Zur Erstellung der Lookup-Tabelle soll hier so vorgegangen werden, dass zunächst der positive Ast der Kennlinie realisiert wird, d.h., Eingangswerte zwischen 2048 (= 0x800) und 4095 (= 0xFFFF) werden auf Ausgangswerte zwischen 128 (= 0x80), entsprechend 0 Volt und 255 (= 0xFF), entsprechend der Maximalaussteuerung abgebildet.

Im Folgenden wird die Codewort-Erzeugung für 12-Bit-Eingangswerte im Bereich 2048 ... 4095, d.h., $2048 + x$ mit $x = 0 \dots 2047$ (= 0x7FF) dargestellt:

<u>Anteil an 2048</u>	<u>Eingangswertebereich (2048+x)</u>	<u>Vz. / Segment / Bits f. lin. Codierung</u>
0 ... 1/64	$0x000 \leq x < 0x020$	1 / 00x ₄ / x ₃ , x ₂ , x ₁ , x ₀
1/64 ... 1/32	$0x020 \leq x < 0x040$	1 / 010 / x ₄ , x ₃ , x ₂ , x ₁
1/32 ... 1/16	$0x040 \leq x < 0x080$	1 / 011 / x ₅ , x ₄ , x ₃ , x ₂
1/16 ... 1/8	$0x080 \leq x < 0x100$	1 / 100 / x ₆ , x ₅ , x ₄ , x ₃
1/8 ... 1/4	$0x100 \leq x < 0x200$	1 / 101 / x ₇ , x ₆ , x ₅ , x ₄
1/4 ... 1/2	$0x200 \leq x < 0x400$	1 / 110 / x ₈ , x ₇ , x ₆ , x ₅
1/2 ... 1	$0x400 \leq x < 0x800$	1 / 111 / x ₉ , x ₈ , x ₇ , x ₆

Unter x_i wird das i -te Bit des 12-Bit-Eingangswortes verstanden (LSB = x_0). Die Ausgangscodeworte ergeben sich aus dem Vorzeichen der Eingangsworte, der Segmentnummer und der linearen Codierung innerhalb eines Segments (Gerade) nach Abbildung 3.2.

Der Eingangswert x kann dabei aus der Laufvariable k (im Bereich $2048 \leq k \leq 4095$) wie folgt gebildet werden: $x = k - 2048$.

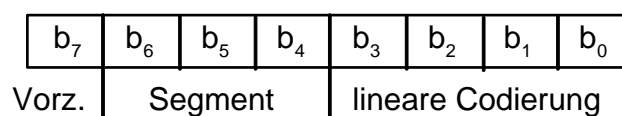


Abbildung 3.2: Codierung eines Kompander-Codeworts

SS 2008

Fakultät für Technik, Studiengänge EIT/TI

Prof. Dr.-Ing. Thorsten Benkner, Dipl.-Ing.(FH) Felix Becker

Negative Eingangsspannungen, d.h., Werte für k im Bereich 0 bis 2047, können durch Spiegelung behandelt werden. Man bildet $x = 2048 - k$ und geht nach der oben beschriebenen Methode vor, wobei das Vorzeichenbit der Codeworte diesmal nicht gesetzt wird.

Man erhält so den in Abbildung 3.1 gezeigten Verlauf der Tabellenwerte.

Die Tabellenwerte können in C besonders effizient mit Hilfe der log. AND/OR-Funktionen und Shift-Operationen berechnet werden. Die Tabelle soll im Variablenfeld `comp[4096]` abgelegt werden (siehe `compand.c`).

3.2 Digitale Expansion

Auf der Empfangsseite wird eine Lookup-Tabelle für den Dekompressionsvorgang benötigt. 8-Bit-Empfangsworte werden in 12-Bit-Ausgangsworte umgerechnet. Die zugehörige Expanderkennlinie ist in Abbildung 3.3 dargestellt.

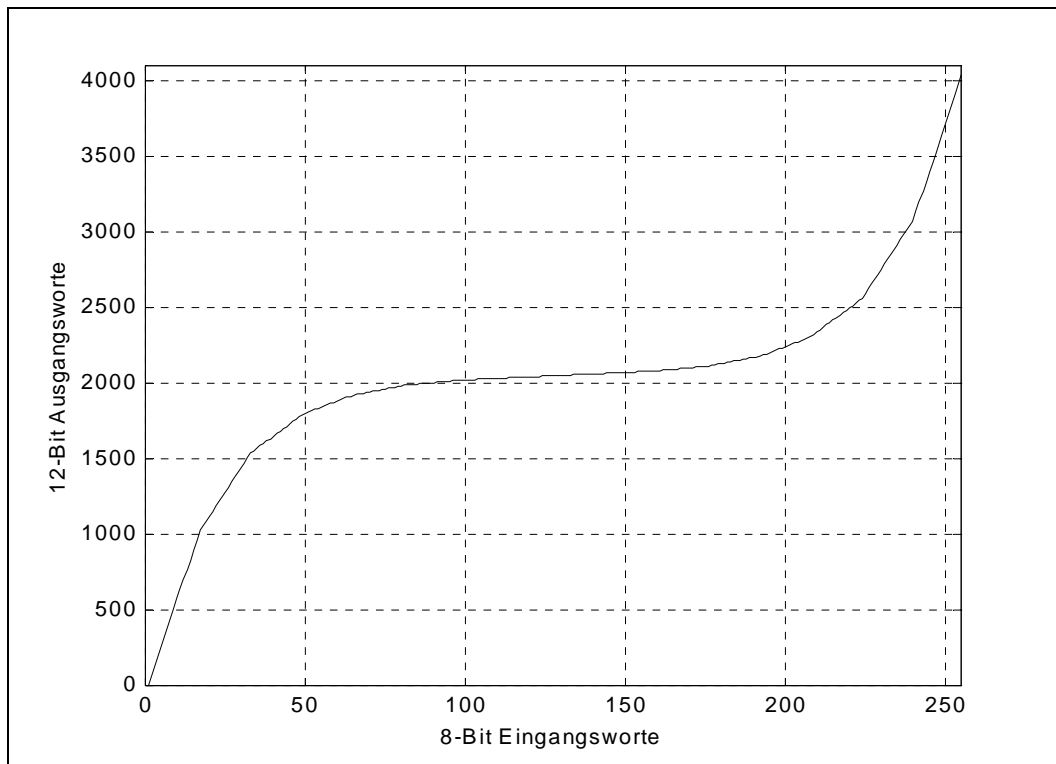


Abbildung 3.3: Kennlinie des Expanders

Am einfachsten wird die Expanderkennlinie direkt aus der Lookup-Tabelle der bereits erstellten Kompressor-kennlinie durch Inversion gewonnen, z.B. durch folgenden Programmteil:

```
/* Expanderkennlinie durch Inversion */
for (k=0; k<4096; k++)
{
    z = comp[k];
    decomp[z] = k;
}
```

Die Expanderkennlinie soll als Lookup-Tabelle im Feld `decomp[256]` abgelegt werden.

SS 2008

Fakultät für Technik, Studiengänge EIT/TI

Prof. Dr.-Ing. Thorsten Benkner, Dipl.-Ing.(FH) Felix Becker

Abbildung 3.4 zeigt die resultierende Kennlinie über einen Kompressions-Expansions-Vorgang. Man erkennt deutlich die unterschiedliche Feinheit der Quantisierung. Große Amplituden werden gröber quantisiert als kleinere, so dass sich über den gesamten Aussteuerungsbereich ein annähernd konstantes Signal-/Quantisierungsrauschleistungsverhältnis einstellt.

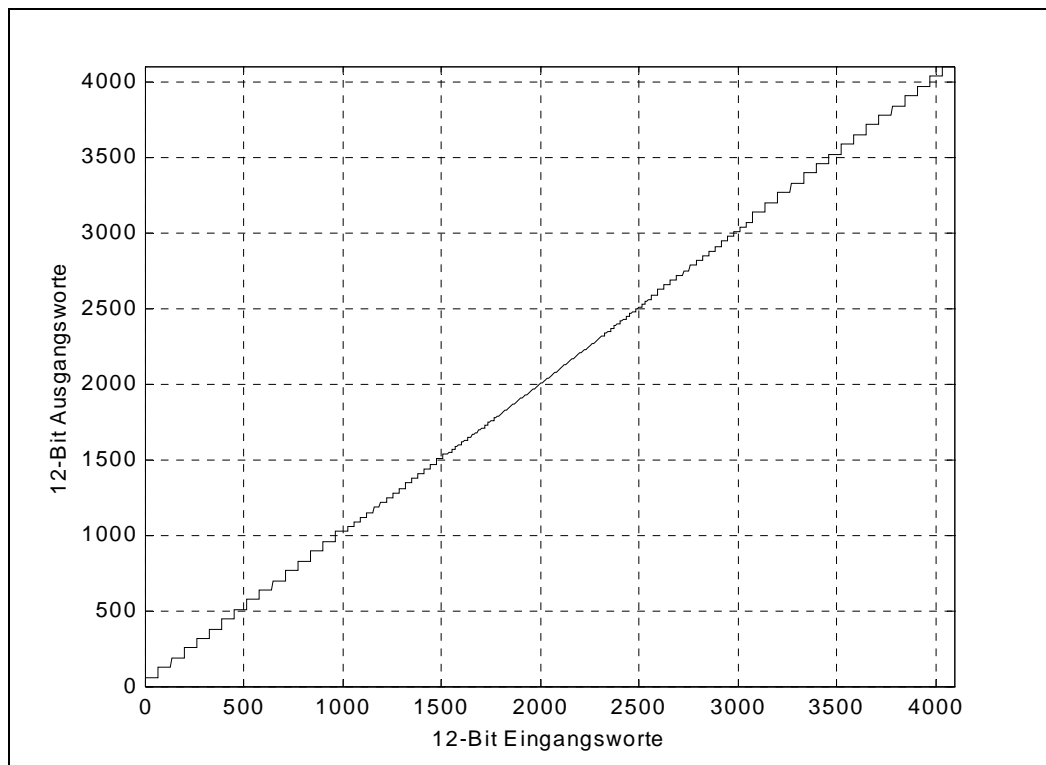


Abbildung 3.4: Resultierende Kennlinie Kompressor - Expander

3.3 Fragen, Vorbereitung und Durchführung

3.1 Fragen:

Welches S/N kann mit einem linearen 8-Bit- und welches mit einem 12-Bit-A/D-Wandler maximal erreicht werden?

Unter welcher Voraussetzung ist dieser maximale Wert möglich?

Welcher Gewinn an Signal-/Quantisierungsrauschleistungsverhältnis stellt sich bei einer Kompondierung mit dem in Abbildung 2.8 gezeigten Verfahren für leise Signale ein?

3.2 Durchführung:

Erstellen Sie mit Code Composer Studio ein neues Projekt für die DSK6711-Plattform. Fügen Sie die zur Verfügung gestellten Quellcode-Dateien (bis auf die Header-Files) dem Projekt hinzu. Nehmen Sie die nötigen Einstellungen bei den Build Options vor (rts-Library, Allocation Order, Heap Size, Stack Size; vgl. Laborversuch 2, Absatz 3.2).

Analysieren Sie die Programm-Module und machen Sie sich die Funktionen klar.

Zunächst soll das Programm als (20-Bit-) "Durchreicher" arbeiten, d.h., Sie müssen die Abtastwerte vom A/D-Wandler einfach wieder über den D/A-Wandler ausgeben. Die

Bitbreite soll dabei allerdings durch die globale Variable `mask` einstellbar (reduzierbar) sein. Die Bearbeitung in `main()` wird nach jedem angekommenen Abtastwert durch die Receive-Interruptroutine angestoßen. (Stereoverarbeitung beachten!!!, d.h. beide Kanäle durchreichen.)

Die Transmit-Interruptroutine erwartet Integer-Werte in den Ausgangsspeichern (rechtsbündig).

Compilieren/Linken Sie das Programm, und laden Sie es auf das DSK.

Testen Sie kurz die korrekte Funktion, indem Sie mit dem Funktionsgenerator `waveGen` des Software-Pakets `AudioTester` einen Sinuston erzeugen und über das DSK übertragen. Testen Sie auch mit passenden Musikbeispielen, die Sie ebenfalls über das DSK übertragen und mit Hilfe des Kopfhörers abhören (zwei geeignete Wave-Dateien finden Sie auf den Labor-PCs über Start -> Vorlagen im Verzeichnis SS-Labor).

Achten Sie jeweils darauf, dass der A/D-Wandler korrekt angesteuert wird, d.h. ermitteln Sie die Aussteuergrenze. Zu Ihrer Hilfe ist dazu in der Receive-Interruptroutine eine Minimalpegelanzeige über die drei LEDs des DSK implementiert. Sie zeigen die Aussteuerung des A/D-Wandlers im Codec an (- 30 dBFS, - 10 dBFS und Peak). Bei korrekter Aussteuerung sollte die `USER_LED3` (Peak-Anzeige) gerade nicht aufleuchten. (Beachten Sie beim Abspielen der beiden Beispiel-Wave-Dateien, dass diese eine hohe Dynamik aufweisen.)

Vermutlich wird Ihnen beim Abhören ein störendes Grundgeräusch (Zirpen) auffallen. Dieses Geräusch wird offenbar durch die permanente Kommunikation der Entwicklungsoberfläche CCS mit dem DSK während der Programmabarbeitung verursacht. Sie können dieses Störgeräusch eliminieren, indem Sie die Programmabarbeitung zunächst anhalten und danach über `Debug` -> `Run Free` wieder starten.

Reduzieren Sie die Wortbreite der durchgereichten Abtastwerte zur Laufzeit mit Hilfe der `WATCH`-Funktion von CCS ausgehend von 20 Bit zunächst auf 12 Bit und dann auf 8 Bit und achten Sie dabei besonders darauf, wie sich das S/N bei reduzierter Auflösung verhält bzw. akustisch bemerkbar macht (bei der Reduktion auf 12 Bit müssen Sie dazu schon genau hinhören).

3.3 Vorbereitung:

Erstellen Sie die Kompressorkennlinie in der Funktion `kennlinie_erstellen()` (siehe Kapitel 3.1 und die Code-Vorgaben in `IProject4\compand.c`).

3.4 Durchführung:

Stellen Sie mittels `View` -> `Graph ...` die Lookup-Tabelle graphisch dar.

(Wenn Sie den lauffähigen Code der Funktion `kennlinie_erstellen()` erst jetzt in die Datei `compand.c` eingefügt haben, müssen Sie die Kennlinie natürlich noch erstellen bevor Sie sie darstellen können. D.h., Sie müssen das Array `comp[4096]` im Speicher durch Starten des Programms füllen. Stoppen Sie den Programmablauf nach kurzer Zeit wieder oder setzen Sie einen Breakpoint hinter den Funktionsaufruf im Hauptprogramm.)

Übernehmen Sie die Grafik in Ihre Versuchsausarbeitung, indem Sie einen Bildchirmausdruck von CCS erzeugen (Druckdaten aus der Zwischenablage dann z.B. in Word einfügen).

Programmieren Sie nun die Expander-Lookup-Tabelle (siehe Kapitel 3.2) und stellen Sie diese ebenfalls graphisch dar. Übernehmen Sie auch diese Grafik in Ihre Versuchsausarbeitung.

3.5 Durchführung:

Programmieren Sie zum Test des Komponders eine Schleife, in der 12-Bit-Worte von 0 bis

SS 2008

Fakultät für Technik, Studiengänge EIT/TI

Prof. Dr.-Ing. Thorsten Benkner, Dipl.-Ing.(FH) Felix Becker

4095 komprimiert und die entsprechenden 8-Bit-Worte sofort wieder expandiert werden. Speichern Sie die Ausgangsworte in einem entsprechenden Variablenfeld ab und stellen Sie auch diese graphisch dar. Es sollte sich eine Gesamt-Übertragungskennlinie wie in Abbildung 3.4 ergeben. Übernehmen Sie die Grafik in Ihre Versuchsausarbeitung.

3.6 Durchführung:

Verändern Sie jetzt Ihr "Durchreicher"-Programm so, dass die A/D-Wandler-Abtastwerte zunächst auf vorzeichenlose 12 Bit reduziert und anschließend im Kompressor auf 8-Bit komprimiert werden (siehe Kapitel 3.1).

Diese Werte sollen nun unverändert auf den Expander gegeben werden, der sie dann wieder auf 12-Bit expandiert.

Die expandierten Werte geben Sie nach der noch nötigen Weiterverarbeitung über den D/A-Wandler aus. (Achten Sie dabei auch darauf, dass der D/A-Wandler richtig angesteuert wird!!!)

Übernehmen Sie die wesentlichen Teile Ihres C-Quellcodes (*compand.c*) ebenfalls in die Versuchsausarbeitung.

Testen Sie nun die korrekte Funktion der Kompondierung mit einem Sinussignal aus dem (Software-) Funktionsgenerator waveGen und mit passenden Musikbeispielen.

Vergleichen Sie das Ergebnis mit dem nicht-kompondierten Fall aus Aufgabenpunkt 3.2. Am direktesten erfolgt der Vergleich, wenn Sie eine Umschaltmöglichkeit zwischen Durchreichen und Kompondieren über den `USER_SW1` vorsehen.

Vergleichen Sie insbesondere das Durchreichen mit 12 Bit Auflösung und das Durchreichen mit 8 Bit Auflösung jeweils mit der Kompondierung.

Beobachten Sie auch, wie sich jeweils das S/N über dem Aussteuerungsbereich, d.h. bei reduzierter A/D-Wandler-Aussteuerung verhält (Messung).

(Die Fragen sind handschriftlich in der Versuchsausarbeitung zu beantworten!)