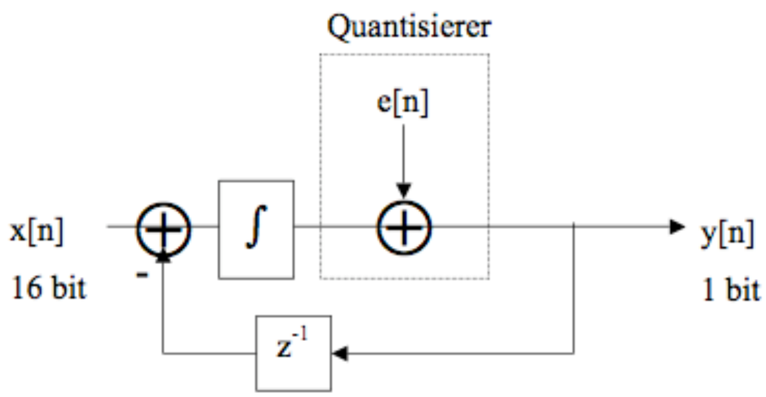
Kommunikationstechnik II – Wintersemester 08/09

Prof. Dr. Stefan Weinzierl

Musterlösung: 5. Aufgabenblatt

Lösung in der Rechenübung am 8.1.2009

1. Aufgabe: Delta- Sigma-Modulation



Der δσ-Modulator ist Grundbaustein von Sigma-Delta-A/D- und -D/A-Wandlern. Es wird die Differenz („Delta") zwischen überabgetastetem Eingangs- und Ausgangssignal integriert („Sigma") und bei sehr hoher Abtastrate auf sehr niedrige Wortbreite quantisiert (s. Abb. 1).

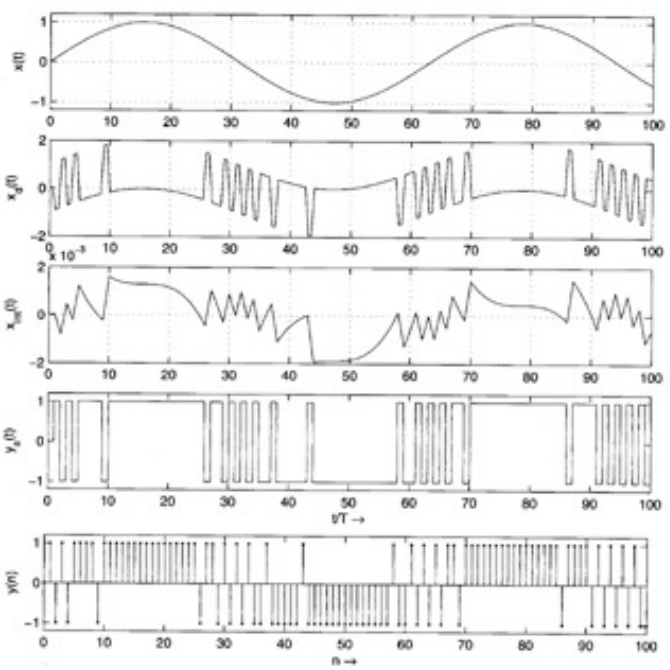


Abb. 1: Delta-Sigma-Modulation (Quelle: Zölzer (2005))

$x[n]$ und $y[n]$ seien Festkommazahlen, normiert auf einen Amplitudenbereich von [-1,1]. Am Quantisierer finde eine Requantisierung auf eine Wortbreite von 1 bit statt.

a) Die Übertragungsfunktion des Integrierers sei $H(z)=\frac{1}{1-z^{-1}}$. Stellen Sie die Differenzengleichung des Systems im Zeitbereich dar.

Differenzengleichung mit $h(n)$ als Impulsantwort des Integrierers:

$$y(n)=\left[x(n)-y(n-1)\right]*h(n)+e(n)$$

In den z-Bereich transformiert:

$$Y(z)=\left[X(z)-z^{-1}Y(z)\right]\cdot H(z)+E(z)=X(z)H(z)-z^{-1}Y(z)H(z)+E(z)$$

$$Y(z)+z^{-1}Y(z)H(z)=X(z)H(z)+E(z)$$

$$Y(z)\left[1+z^{-1}H(z)\right]=X(z)H(z)+E(z)=\frac{H(z)}{1+z^{-1}H(z)}X(z)+\frac{1}{1+z^{-1}H(z)}E(z)$$

mit $H(z)=\frac{1}{1-z^{-1}}$:

$$\frac{\frac{1}{1-z^{-1}}}{1+z^{-1}\frac{1}{1-z^{-1}}}X(z)+\frac{1}{1+z^{-1}\frac{1}{1-z^{-1}}}E(z)=\frac{\frac{1}{1-z^{-1}}}{\frac{1-z^{-1}}{1-z^{-1}}+\frac{z^{-1}}{1-z^{-1}}}X(z)+\frac{1}{\frac{1-z^{-1}}{1-z^{-1}}+\frac{z^{-1}}{1-z^{-1}}}E(z)$$

$$=\frac{\frac{1}{1-z^{-1}}}{\frac{1}{1-z^{-1}}}X(z)+\frac{1}{\frac{1}{1-z^{-1}}}E(z)=X(z)+(1-z^{-1})E(z)$$

Dies entspricht Noiseshaping erster Ordnung!

Zurück transformiert in den Zeitbereich:

$$y(n)=x(n)-e(n-1)+e(n)$$

b) Berechnen Sie den Ausgang $y[n]$ für ein Eingangssignal $x[n]$ mit der konstanten Amplitude 0.7 und stellen Sie Eingangs- und Ausgangssignal über eine Dauer von 20 Samples dar.

c) Berechnen Sie den Ausgang $y[n]$ für ein voll ausgesteuertes Sinussignal als Eingang $x[n]$ mit einer Periode von 20 Samples und stellen Sie Eingangs- und Ausgangssignal über eine Periodendauer dar.

Um das Abtasttheorem zu erfüllen, muss die Anzahl der Samples innerhalb einer Periode mehr als 2 betragen, da $f_s > 2\cdot f_{max}$. Eine konstante Eingangsamplitude ebenso wie eine Periode eines Sinussignals sind also mit 20 samples überabgetastet.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Musterloesung zum 5. Aufgabenblatt                                        %
%                                                                           %
```

```
%-----------------------------------------------------------------------------%
% Aufgabe 1: Delta-Sigma-Modulation

%%%%%%%% b) und c) %%%%%%%%

% Differenzengleichung:
% y(n) = x(n) + e(n) - e(n-1)

% Aufgabe b)

% Erzeuge mit 16 bit quantisiertes Eingangssignal:
x1 = ones(1,20)*0.7;
q = 2/2^16;
x1 = quant(x1,q);

% Eine zusätzliche Null stellt den Registerzustand (noch ohne den Anteil
% des Rückkopplungszweigs) im Verzögerungsglied dar, bevor das
% Eingangssignal am Quantisierer angelangt ist:
x1 = [0 x1];

% Initialisiere einen Fehlervektor
e = zeros(1,length(x1));
% Berechne Ausgangssignal
for n = 2:21;
    % bestimme den aktuellen Zustand vor dem Quantisierer:
    zustand = x1(n) - e(n-1);

    % 1-Bit-Quantisierung des Signals,
    % hierbei entsteht auch der Quantisierungsfehler e(n)
    if zustand >= 0
        y1(n-1) = 1;
    else
        y1(n-1) = -1;
    end

    %Bestimme den Quantisierungsfehler e(n)
    e(n) = y1(n-1) - zustand;
end


% Aufgabe c)

% Zeitvektor für eine Sinus-Periode
t = 0:1/20:1-1/20;

% Erzeuge mit 16 bit quantisiertes Eingangssignal
x2 = sin(2*pi*t);
Q = 2/2^16;
x2 = quant(x2,Q);

x2 = [0 x2];
e = zeros(1,length(x2));

for n = 2:21;

    zustand = x2(n) - e(n-1);

    if zustand >= 0
        y2(n-1) = 1;
    else
```

```
        y2(n-1) = -1;
    end

    e(n) = y2(n-1) - zustand;
end


% Plots
figure
subplot(2,2,1), stem(x1), title('Eingangsignal Aufg b)')
xlabel('samples'), ylabel('Amplitude'), axis([0 20 -1 1]), grid on
subplot(2,2,2), stem(y1), title('Ausgangssignal Aufg b)')
xlabel('samples'), ylabel('Amplitude'), grid on
subplot(2,2,3), stem(x2), title('Eingangsignal Aufg c)')
xlabel('samples'), ylabel('Amplitude'), axis([0 20 -1 1]), grid on
subplot(2,2,4), stem(y2), title('Ausgangssignal Aufg c)')
xlabel('samples'), ylabel('Amplitude'), grid on
```

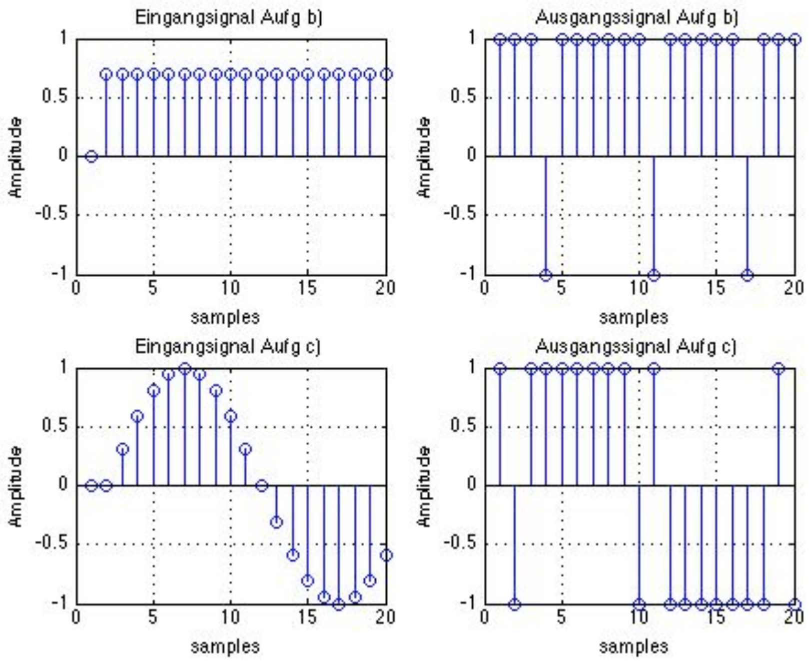


Abb. 2: Input und Output der DSM

Berechnung der ersten 6 samples aus Aufgabe b):

$$y(0) = [x(0) - e(-1)]_Q = [0{,}7 - 0]_Q = [0{,}7]_Q = 1$$

$$e(0) = 1 - 0{,}7 = 0{,}3$$

$$y(1) = [x(1) - e(0)]_Q = [0{,}7 - 0{,}3]_Q = [0{,}4]_Q = 1$$

$$e(1) = 1 - 0{,}4 = 0{,}6$$

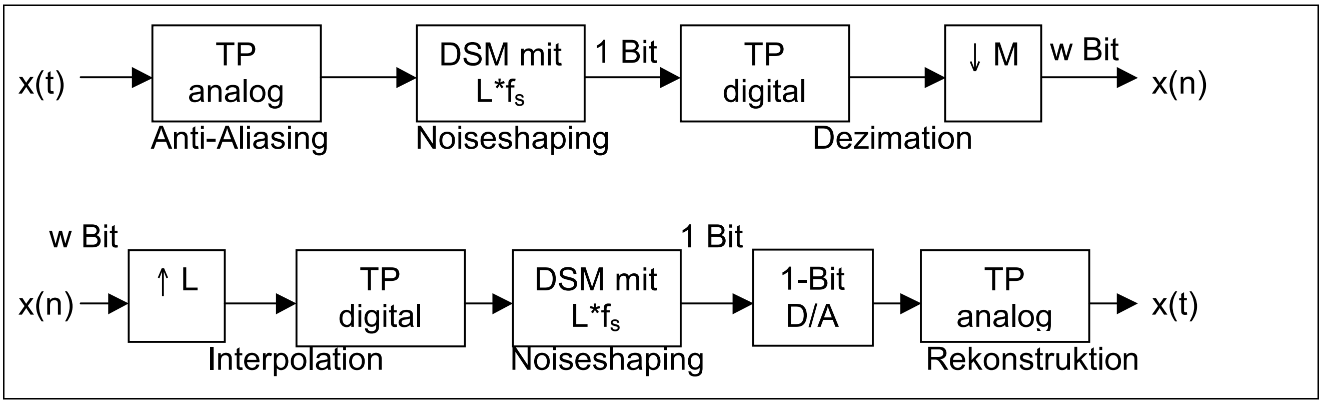
$$y(2) = [x(2) - e(1)]_Q = [0{,}7 - 0{,}6]_Q = [0{,}1]_Q = 1$$
$$e(2) = 1 - 0{,}1 = 0{,}9$$

$$y(3) = [x(3) - e(2)]_Q = [0{,}7 - 0{,}9]_Q = [-0{,}2]_Q = -1$$
$$e(3) = -1 + 0{,}2 = -0{,}8$$

$$y(4) = [x(4) - e(3)]_Q = [0{,}7 + 0{,}8]_Q = [1{,}5]_Q = 1$$
$$e(4) = 1 - 1{,}5 = -0{,}5$$

$$y(5) = [x(5) - e(4)]_Q = [0{,}7 + 0{,}5]_Q = [1{,}2]_Q = 1$$
$$e(5) = 1 - 1{,}2 = -0{,}2$$

Um einen kompletten A/D- bzw. D/A-Wandler zu realisieren benötigt der δσ-Modulator noch weitere Module



Blockschaltbild Delta-Sigma-Wandler; oben: A/D; unten: D/A

Überabtastender Delta-Sigma-A/D-Wandler:

- im analogen Fall ist vor dem δσ-Modulator lediglich eine Nyquist-Bandbegrenzung notwendig, der δσ-Modulator arbeitet bis zur 1-Bit-Quantisierungsstufe analog
- das Signal liegt nach der δσ-Modulation L-fach überabgetastet im 1-Bit-Format vor
- um das Signal in ein 16-Bit-, 44-kHz Signal zu überführen, muss Dezimation (siehe Aufgabenblatt 4, Aufgabe 1) stattfinden. TP realisierbar durch FIR-Filter → wirkt wie gleitende Durchschnittsbildung, Ausgangssignal enthält dadurch wieder mehr Amplitudenstufen (~Wortbreitenerhöhung)
- Senkung der Abtastrate, durch Verwerfen von jeweils M-1 Samples

Überabtastender Delta-Sigma-D/A-Wandler:

- Delta-Sigma-D/A Wandler benötigt interpoliertes Signal (siehe Aufgabenblatt 4, Aufgabe 1), am Eingang liegt also ein diskretes, überabgetastetes und bandbeschränktes Signal
- Rekonstruktion des 1-Bit Signals durch analogen Tiefpass bei $f_{nyquist}$

Vorteile des δσ-Modulators:
Signal liegt L-fach überabgetastet und inklusive Noiseshaping vor
→ Rauschleistung des Quantisierungsfehlers durch Noiseshaping **und** Überabtastung spektral günstig verteilt
→ beim δσ-A/D-Wandler verringerter Anspruch an analoges Antialiasing-Filter, beim δσ-D/A-Wandler an analoges Rekonstruktionsfilter (dadurch z.B. besseres Phasenverhalten)

2. Aufgabe: Output-Kodewörter und Messsignal

a) Wie viele digitale Output-Kodewörter kann ein 16-Bit- und ein 20-Bit-Wandler erzeugen?

Es gilt: $Q = 2^n$
Ein 16-Bit-Wandler kann demnach 2^{16} = 65’536 verschiedene Kodewörter erzeugen, bei einem 20-Bit-Wandler sind es 2^{20} = 1'048’576.

b) Wie viele dieser Output-Kodewörter werden innerhalb von 80 ms bei der A/D-Wandlung mit f_s = 48 kHz
- von einem voll ausgesteuerten Sinussignal mit f = 1 kHz?
- von einem voll ausgesteuerten Sinussignal mit f = 997 Hz?

Es wird bei beiden Signalen innerhalb von 80 ms dieselbe Anzahl von Kodewörtern ausgelöst. Die Abtastfrequenz wird nicht von der Signalfrequenz beeinflusst, sodass innerhalb einer festgelegten Zeit gleich viele Abtastzeitpunkte auftreten. Bei der gegebenen Abtastfrequenz sind dies 48'000 Abtastzeitpunkte pro Sekunde, also treten in 80 ms

48'000 Hz * 0.08 s = 3840 Abtastzeitpunkte bzw. Kodewörter

auf. Allerdings sind dies bei 997 Hz Signalfrequenz *verschiedene* Kodewörter und bei 1 kHz Signalfrequenz nicht (siehe Aufgabe c)). In beiden Fällen werden jedoch nicht alle, durch Wortbreite theoretisch möglichen Amplitudenstufen erreicht (vgl. Aufgabe a)). Bei längerer Signaldauer könnte dies verbessert werden.

c) Erzeugen Sie die zugehörigen Abtastfolgen in Matlab und lassen Sie sich mit der Funktion `hist` die Häufigkeitsverteilung der einzelnen Kodes während einer Signaldauer von 80 ms für einen 16-Bit-Wandler anzeigen.

```
% Aufgabe 2: Output-Kodewoerter und Messsignal

%%%%%%%% c) %%%%%%%%

% Sinus 1000 Hz und 997 Hz, 80ms lang
fs = 48000;
f1 = 1000;
f2 = 997;
t = 0:1/fs:0.08-1/fs;
sinus_1000 = sin(2*pi*f1*t);
```

```
sinus_997 = sin(2*pi*f2*t);

% Quantisierung mit 16 bit
w = 16;                          % Wortbreite
q = 2^-(w-1);    % Quantisierungsintervall
sinus_1000q = quant(sinus_1000,q);
sinus_997q = quant(sinus_997,q);

%Haeufigkeitsverteilung in 2^w bins
n = 2^w;
anzahl_1000= hist(sinus_1000q,n);
anzahl_997= hist(sinus_997q,n);
% Vektor mit den Haufigkeiten ≠ 0
z_1000 = find(anzahl_1000);
z_997 = find(anzahl_997);
% seine Laenge gibt an, wieviele unterschiedliche Kodewoerter vorkommen.
length(z_1000)
length(z_997)

figure
hist(sinus_1000q,n), title('Haeufigkeitsverteilung Sinus 1000 Hz, 16 bit')
axis([-1 1 0 180])
xlabel('Quantisierungsstufen'), ylabel('Haeufigkeit')

figure
hist(sinus_997q,n), title('Haeufigkeitsverteilung Sinus 997 Hz, 16 bit')
xlabel('Quantisierungsstufen'), ylabel('Haeufigkeit')
```

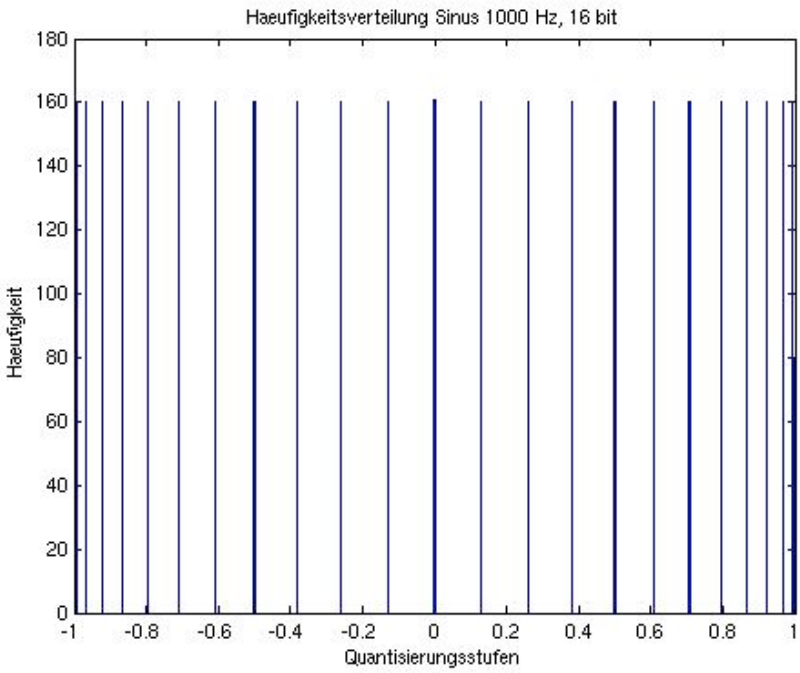


Abb. 3: Häufigkeitsverteilung eines 1-kHz-Sinussignals, mit 48 kHz abgetastet und mit 16 bit quantisiert

Die Häufigkeitsverteilung des 1-kHz-Sinussignals ist vollständig symmetrisch. Die Werte, die in der ersten Periode abgetastet wurden, wiederholen sich exakt in jeder der folgenden Perioden, weil die Abtastfrequenz ein ganzzahliges Vielfaches der Signalfrequenz ist:

$\frac{f_{\text{Abtast}}}{f_{\text{Signal}}} = 48$. Die Periodendauer des Signals T_{Signal} beträgt 0.001 s, die Anzahl der Perioden innerhalb 80 ms beträgt also $\frac{\text{Signaldauer}}{T_{\text{Signal}}} = 80$. In jeder Periode treten dieselben 48 Kodewörter auf.
Da bei einer Sinusschwingung jeder Wert außer 1 und -1 im Laufe einer Periode zwei Mal vorkommt, ergeben sich faktisch sogar noch weniger verschiedene Kodewörter.
48 - 2 = 46 Kodewörter ohne Maximum und Minimum
46/2 +2 = 23 + 2 *verschiedene* Werte (inkl. Maximum und Minimum)
Die Häufigkeit der Kodewörter beträgt 160 für alle Werte. Für 1 und -1 ist die Häufigkeit 80, da diese beiden Werte nur je ein Mal pro Periode vorkommen.

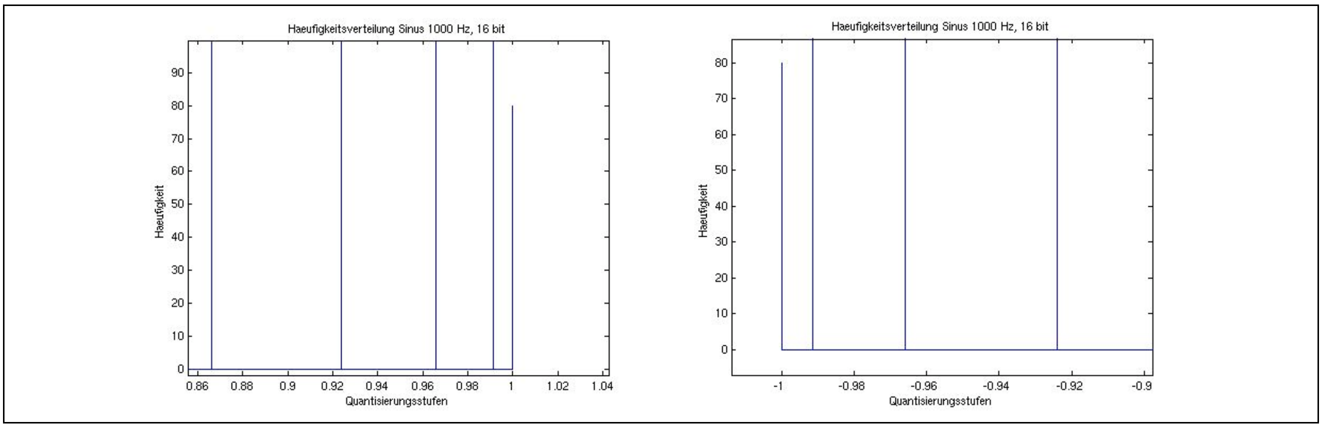


Abb. 4: Detailansicht von Abb. 3; links: um Amplitudenwert 1; rechts: um Amplitudenwert -1

Die Häufigkeitsverteilung des 997-Hz-Signals ergibt ein annähernd gleichverteiltes Bild. Hier treten fast alle Werte nur ein Mal oder gar nicht auf. Dies lässt sich dadurch begründen, dass im vorliegenden Fall die Abtastfrequenz kein ganzzahliges Vielfaches der Signalfrequenz mehr ist: $\frac{f_{\text{Abtast}}}{f_{\text{Signal}}} = 48{,}144$. Dies führt dazu, dass sich die Abtastwerte nicht in jeder Periode wiederholen, sondern immer neue Kodewörter generiert werden.

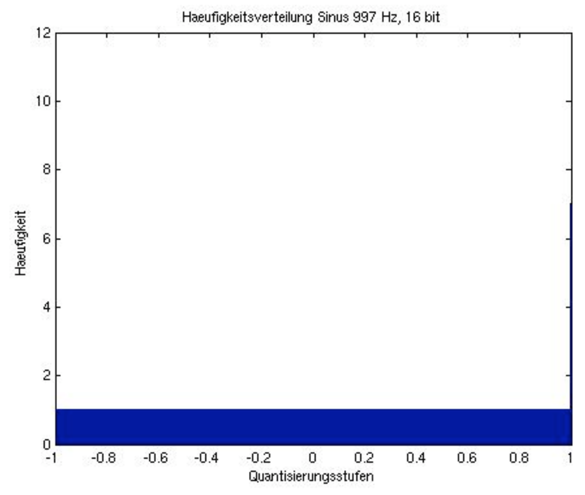


Abb. 5: Häufigkeitsverteilung eines 997-Hz-Sinussignals, mit 48 kHz abgetastet und mit 16 bit quantisiert

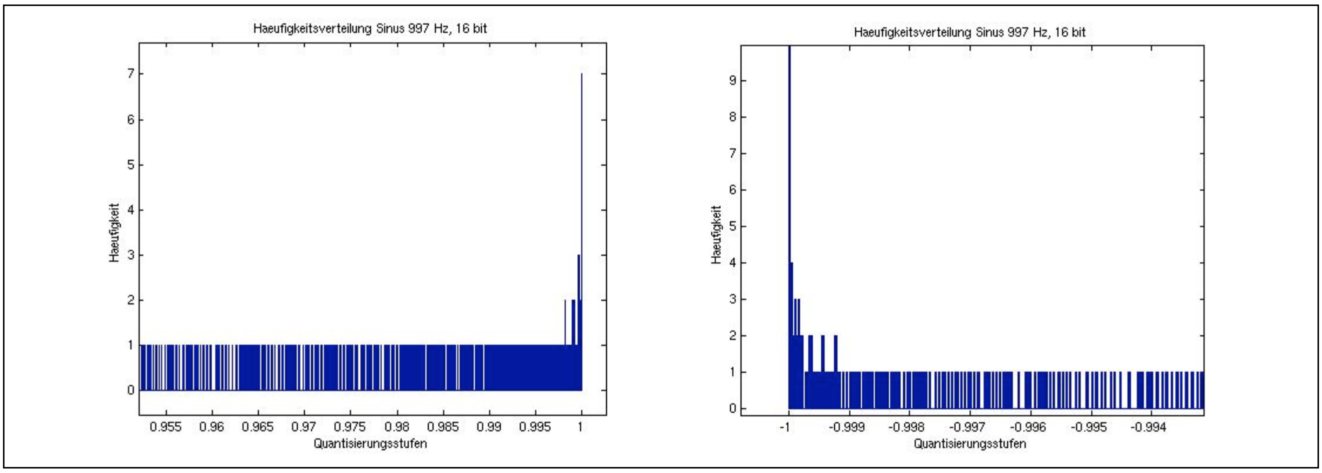


Abb. 6: Detailansicht von Abb. 5; links: um Amplitudenwert 1; rechts: um Amplitudenwert -1

Die größere Häufigkeit um die Werte 1 und -1 lässt sich dadurch erklären, dass sich das Signal um diese Amplitudenstufen herum „langsamer ändert“, sodass dort mehr Abtastzeitpunkte hinfallen.

d) Erklären Sie den Sinn dieser „ungeraden“ Frequenz als Festschreibung für das Messsignal im AES17-Standard.

Diese Frequenz wird deshalb gewählt, weil, wie oben gezeigt werden konnte, wesentlich mehr verschiedene Kodewörter erzeugt werden, deren Häufigkeiten annähernd gleich verteilt sind. Bei den Messungen nach dem AES17-Standard ist es von Bedeutung, dass eine möglichst große Bandbreite von Kodewörtern getestet wird, da es entscheidend ist, eventuell auftretende Fehler mit zu messen. Wird das Messsignal so gewählt, dass die Abtastfrequenz ein ganzzahliges Vielfaches ist, so ergibt sich ein verfälschtes Messergebnis. Es werden in so einem Fall nur $\frac{f_{\text{Abtast}}}{f_{\text{Signal}}}$ Messwerte berücksichtigt, was keinesfalls repräsentativ ist für die hier 2^n möglichen Kodewörter.