Musterlösung: 2. Dezember 2013, 13:41

1 Delta-Sigma-Modulation

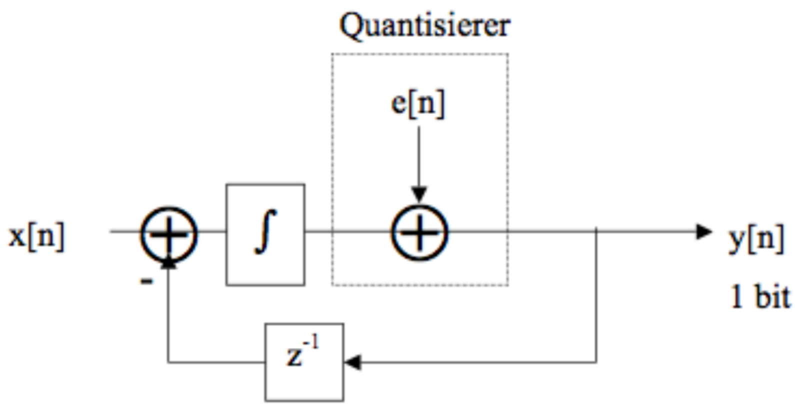


Abbildung 1: Blockschaltbild eines Delta-Sigma-Modulators

Der in Abb. 1 dargestellte Delta-Sigma-Modulator nimmt eine 1-Bit Quantisierung der bereits abgetasteten Zahlenfolge $x[n]$ vor. Dabei sind $x[n]$ und $y[n]$ auf einen Amplitudenbereich von $[-1,\ 1]$ normiert.

a) Erläutern Sie kurz die Funktionsweise eines Delta Sigma Modulators.
Lösung: Der $\delta\sigma$-Modulator ist Grundbaustein von Sigma-Delta-A/D- und -D/A-Wandlern. Es wird die Differenz („Delta") zwischen Eingangs- und Ausgangssignal integriert („Sigma") und bei sehr hoher Abtastrate auf sehr niedrige Wortbreite quantisiert. Dabei wird der Quantisierungsfehler Spektral geformt. Durch eine Abtastung des Eingangssignals mit dem L-fachen der gewünschten Abtastrate lassen sich trotz der niedrigen Wortbreite sehr gute SNRs realisieren.

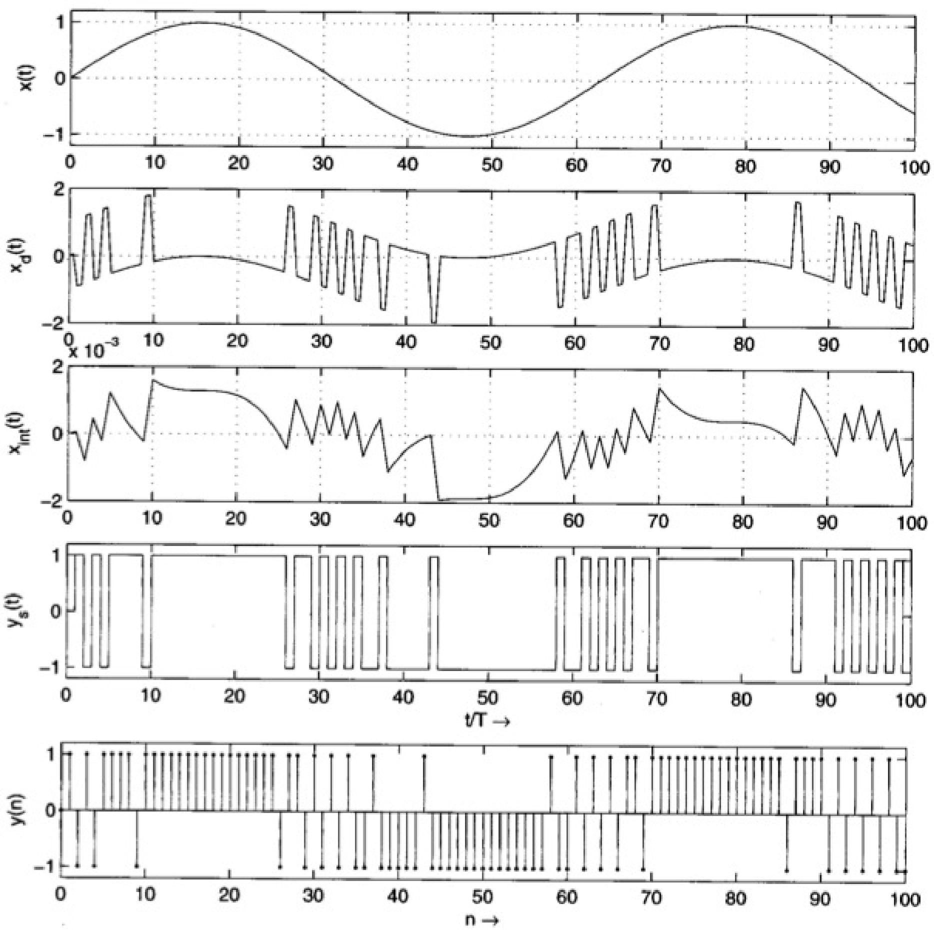
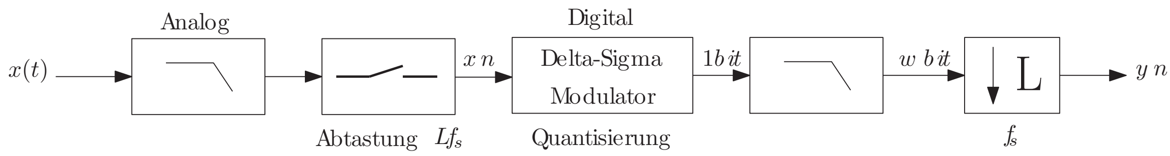


Abbildung 2: Blockschaltbild eines Delta-Sigma-Modulation (Quelle Zölzer (2005): Digitale Audiosignalverarbeitung)

b) Welche Filter werden benötigt, wenn eine A/D-Wandlung mit Hilfe eines Delta Sigma Modulators realisiert werden soll? Zeichnen Sie das dazugehörige Blockschaltbild.
Lösung: Es wird ein analoger Tiefpass benötigt um Signalanteile oberhalb der Nyquist-Frequenz zu unterdrücken und ein digitaler Tiefpass, um den 1 bit-Stream in ein Signal höherer Wortbreite zu wandeln.



c) Die Übertragungsfunktion des Integrieres sei $H(z) = 1/(1 - z^{-1})$. Überlegen Sie sich anhand des Pol-Nullstellendiagrams, um was für eine Art Filter es sich handelt und geben sie die Differenzengleichung des Integrierers an?
Lösung: Um abschätzen zu können, um was für eine Art Filter es sich handelt werden Pol- und Nullstellen berechnet und im Pol- Nullstellen-Diagramm dargestellt.

$$H(z) = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}$$
$$\Rightarrow n_1 = 0 \qquad und \qquad p_1 = 1$$

Wegen der Polstelle bei 1 (was der Frequenz $f = 0$ Hz entspricht) handelt es sich um eine Art

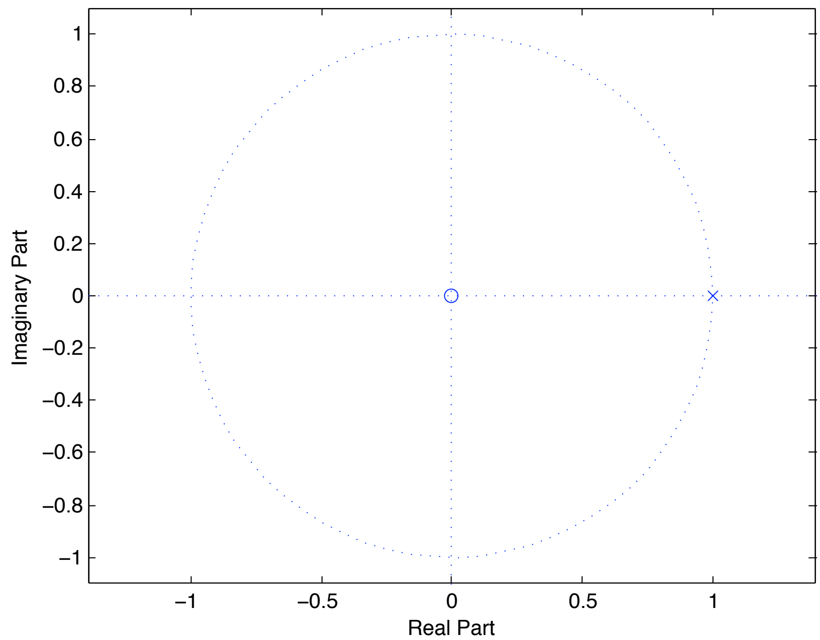


Abbildung 3: Pol-Nullstellen-Diagramm des Integrierers

Tiefpass. Daher auch die Bezeichnung Integrierer. Die Übertragungsfuntion kann einfach mit Matlab geplottet werden (Siehe Abb. 4).

```
%% c)
[h, f] = freqz([1 0], [1 -1], 2^13, 44100);
```

```
semilogx(f/1000, 20*log10(abs(h)))
axis([.01 20 0 60])
grid on
set(gca, 'xTickLabel', {'10', '100', '1k', '10k', '20k'})
xlabel 'f in Hz'
ylabel '|H| in dB'
```

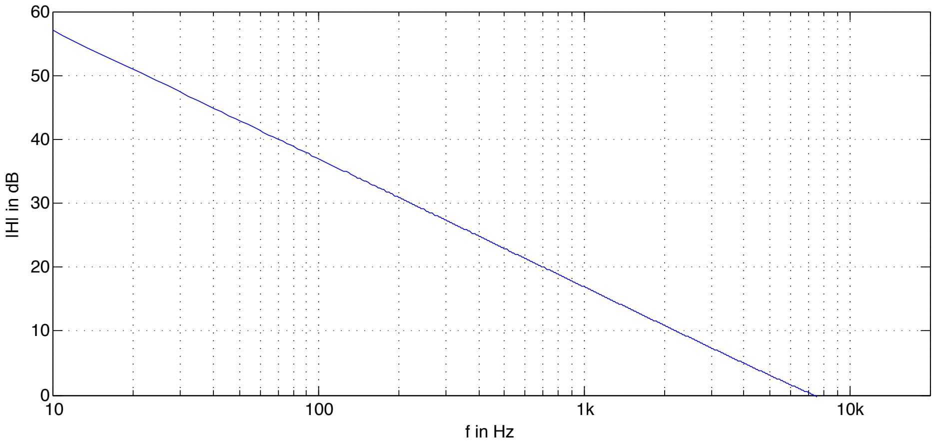


Abbildung 4: Betragsfrequenzgang des Integrierers

Da sich die Polstelle des Integrierers auf dem Einheitskreis befindet handelt es sich um ein instabiles System. Dies wird amhand der Differenzengleichung deutlich. Aus

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{n=0}^{N} b_n\, z^{-n}}{a_0 + \sum_{m=1}^{M} a_m\, z^{-m}}$$

folgt:

$$Y(z)\left(a_0 + \sum_{m=1}^{M} a_m\, z^{-m}\right) = X(z) \sum_{n=0}^{N} b_n\, z^{-n}$$
$$Y(z)\, a_0 = X(z) \sum_{n=0}^{N} b_n\, z^{-n} - Y(z) \sum_{m=1}^{M} a_m\, z^{-m}$$

und schließlich:

$$y[n] = x[n] + y[n-1]$$

Zum aktuellen Eingangswert wird also immer der letzte Ausgangswert addiert. Da der Integrierer jedoch nur Teil des Gesamtsystems Delta-Sigma-Modulator ist, kann anhand dessen noch keine Aussage über die Stabilität des Gesamtsystems gemacht werden.

d) Stellen Sie die Differenzengleichung des Sigma-Delta-Modulators im Zeitbereich dar.
Lösung: Differenzengleichung mit $h[n]$ als Impulsantwort des Integrierers:

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

In den z-Bereich transformiert:

$$\begin{aligned}
Y(z) &= [X(z) - Y(z)z^{-1}] \cdot H(z) + E(z)\\
&= X(z)H(z) - Y(z)z^{-1}H(z) + E(z)\\
Y(z) + Y(z)z^{-1}H(z) &= X(z)H(z) + E(z)\\
Y(z)(1 + z^{-1}H(z)) &= X(z)H(z) + E(z)\\
Y(z) &= \frac{H(z)}{1 + z^{-1}H(z)}X(z) + \frac{1}{1 + z^{-1}H(z)}E(z)
\end{aligned}$$

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

$$\begin{aligned}
Y(z) &= \frac{\frac{1}{1-z^{-1}}}{1 + z^{-1}\frac{1}{1-z^{-1}}}X(z) + \frac{1}{1 + z^{-1}\frac{1}{1-z^{-1}}}E(z)\\
&= \frac{\frac{1}{1-z^{-1}}}{\frac{1-z^{-1}}{1-z^{-1}} + \frac{z^{-1}}{1-z^{-1}}}X(z) + \frac{1}{\frac{1-z^{-1}}{1-z^{-1}} + \frac{z^{-1}}{1-z^{-1}}}E(z)\\
&= \frac{\frac{1}{1-z^{-1}}}{\frac{1}{1-z^{-1}}}X(z) + \frac{1}{\frac{1}{1-z^{-1}}}E(z)\\
&= X(z) + E(z)(1 - z^{-1})
\end{aligned}$$

Das entspricht Noiseshaping erster Ordnung! Zurücktransformiert in den Zeitbereich ergibt sich:

$$y[n] = x[n] + e[n] - e[n-1]$$

e) Berechnen Sie den Ausgang $y[n]$ für ein 5 Samples langes Eingangssignal $x[n]$ mit der konstanten Amplitude 0.7.
Lösung: Das $e[n]$ in der Differenzengleichung des Delta-Sigma-Modulators kann auch als Quantisierung Q interpretiert werden. So lassen sich die gesuchten Ausgangswerte leicht berechnen. Dabei werden Werte < 0 zu -1 und Werte ≥ 0 zu 1 Quantisiert.

$$\begin{aligned}
&y[n] = x[n] - e[n-1] + e[n] = (x[n] - e[n-1])_Q \qquad \textit{und}\\
&e[n] = y[n] - (x[n] - e[n-1])
\end{aligned}$$

Berechnung der Ausgangswerte:

$$\begin{aligned}
&y[0] = (x[0] - e[-1])_Q = (0.7)_Q = 1\\
&e[0] = y[0] - (x[0] - e[-1]) = 1 - 0.7 = 0.3\\
\\
&y[1] = (x[1] - e[0])_Q = (0.7 - 0.3)_Q = 1\\
&e[1] = y[1] - (x[1] - e[0]) = 1 - 0.4 = 0.6\\
\\
&y[2] = (x[2] - e[1])_Q = (0.7 - 0.6)_Q = 1\\
&e[2] = y[2] - (x[2] - e[1]) = 1 - 0.1 = 0.9\\
\\
&y[3] = (x[3] - e[2])_Q = (0.7 - 0.9)_Q = -1\\
&e[3] = y[3] - (x[3] - e[2]) = -1 + 0.2 = -0.8\\
\\
&y[4] = (x[4] - e[3])_Q = (0.7 + 0.8)_Q = 1\\
&e[4] = y[4] - (x[4] - e[3]) = 1 - 1.5 = -0.5
\end{aligned}$$

f) Implementieren Sie die Differenzengleichung aus d) in einer Matlabfunktion

$$[y1bit, e] = dsm(x)$$

Testen Sie die Funktion mit einem voll ausgesteuertem Sinussignal mit einer Periodendauer von 20 Samples, sowie dem Signal aus e) und stellen Sie Eingangs- und Ausgangssignal dar.
Lösung: Der Code für die Funktion dsm.m befindet sich hinter Aufgabenteil g.

```
%% f)
% Erzeuge konstantes Eingangssignal
x1 = ones(1,20)*.7;

% Erzeuge Sinus
% Zeitvektor f?r eine Sinus-Periode
t = 0:1/20:1-1/20;

% Erzeuge mit 16 bit quantisiertes Eingangssignal
x2 = sin(2*pi*t);

[y1, e1] = dsm(x1);
[y2, e2] = dsm(x2);

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

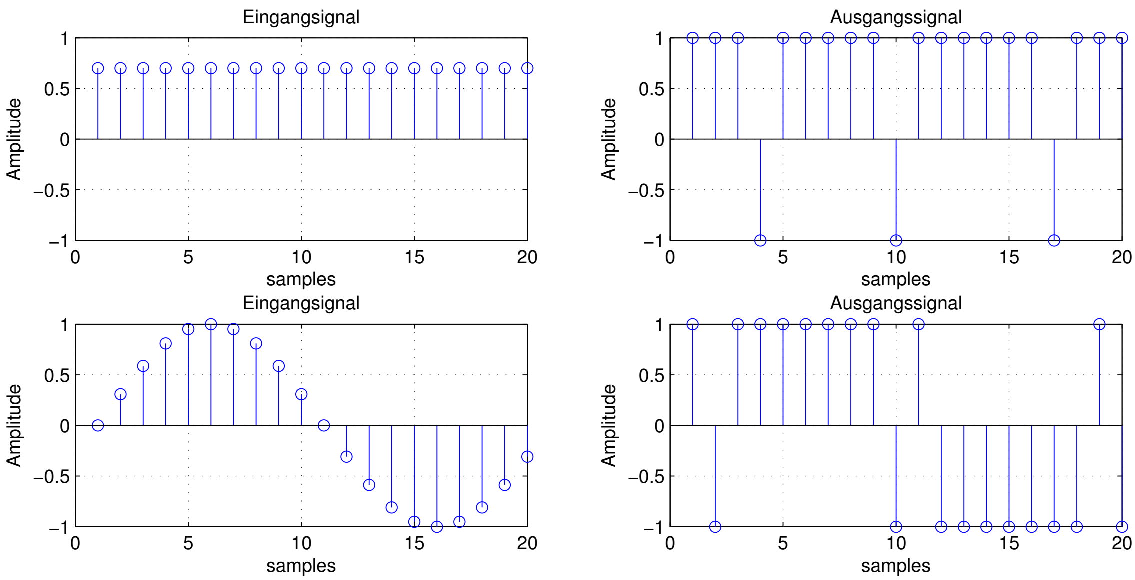


Abbildung 5: Eingangssignal und 1 Bit-Ausgangssignal des Delta-Sigma-Modulators

Abb. 5 zeigt Eingangssignale auf der linken und die Ausgangssignale vor der digitalen Tiefpassfilterung und Unterabtastung auf der rechten Seite.

g) Erweitern Sie die Matlabfunktion *dsm.m* um ein Dezimationsfilter (Zölzer: Digitale Audiosignalverarbeitung, 2005 Gl. 3.30) mit anschließender Unterabtastung. Testen Sie die Funktion mit einem geeigneten Signal und plotten Sie dessen Spektrum.
Lösung: Das Dezimationsfilter ist gegeben durch

$$H_{TP}(z) = \frac{1}{L} \cdot \frac{1 - z^{-L}}{1 - z^{-1}}$$

Als Eingangssignal wird ein Sinus mit $f = 200$ Hz bei einer Abtastfrequenz von $f_s = 8$ kHz und ein Überabtastungsfaktor von $L = 20$ gewählt.

```
%% g)
L  = 20;
fs = 8000;
f  = 200;
n  = 1:L*fs-1;

x3 = sin(n*2*pi*f/(L*fs));

[y3_1bit, e3, y3] = dsm(x3, L);

Y3 = abs(fft(y3)/max(fft(y3)));
f  = 0:fs/length(y3):fs-fs/length(y3);

% Unterabgetastetes Eingagnssignal
soundsc(x3(1:L:end), fs)
% Ausgangssignal
soundsc(y3, fs)


%% plot
figure
    plot(f, 20*log10(Y3))
    xlabel('f in Hz'); ylabel('dB Fs'); title('Spektrum nach Delta-Sigma A/D-Wandlung')
```

Bei der Hörkontrolle des Delta-Sigma gewandelten Signals sind deutliche Verzerrungsprodukte zu erkennen, die sich auch im Spektrum zeigen (Abb. 6).
Die bei ca. -30 dB befindlichen Obertöne der 200 Hz Schwingung entstehen durch die geringe Sperrdämpfung des verwendeten Dezimationsfilter (siehe Abb. 7). Um diese zu verbessern werden üblicherweise mehrere Filter kaskadiert.

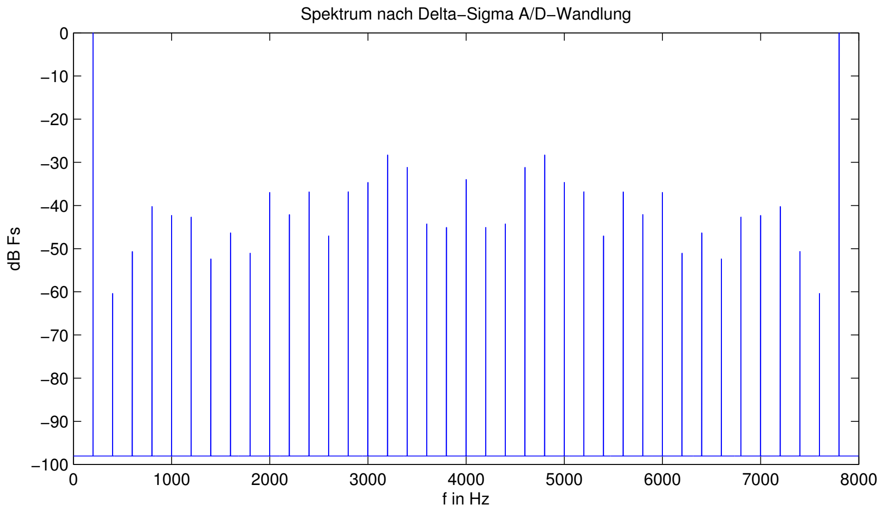


Abbildung 6: Spektrum des unterabgetasteten Ausgangssignals

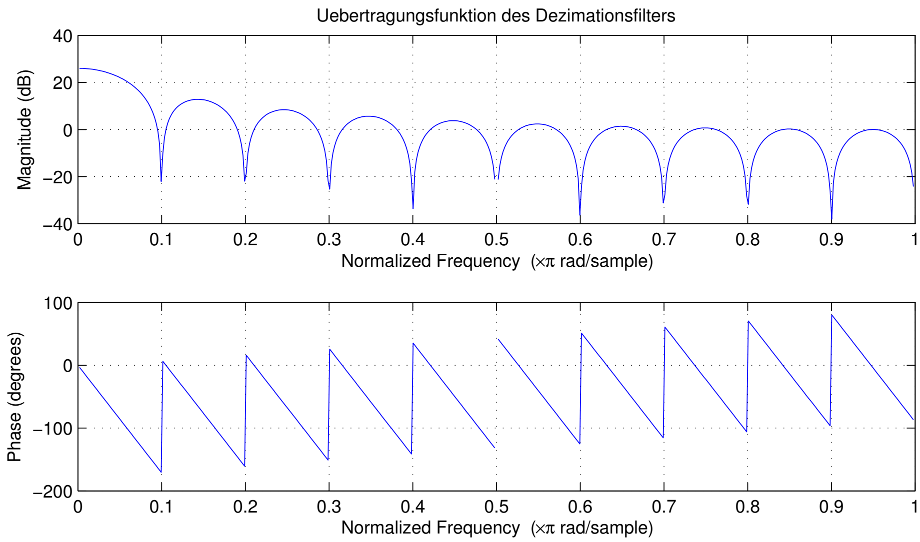


Abbildung 7: Frequenzgang des Dezimationsfilters

Funktion *dsm.m*:

```
function [y1bit, e, y] = dsm(x, L)

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

% Initialisiere Fehlervektor und Ausgabevektor
y1bit = zeros(size(x));
e = zeros(size(x));
e_buf = 0;

% Berechne 1 Bit Ausgangssignal
for n = 1:length(x);
    % bestimme den aktuellen Zustand vor dem Quantisierer:
    zustand = x(n)-e_buf;

    % 1-Bit-Quantisierung des Signals,
    % hierbei entsteht auch der Quantisierungsfehler e(n)
    if zustand >= 0
        y1bit(n) = 1;
    else
        y1bit(n) = -1;
    end

    %Bestimme den Quantisierungsfehler e(n)
    e_buf = y1bit(n) - zustand;
    e(n)  = e_buf;
end

% Dezimation und Unterabtastung
if nargin > 1

    % H(z) = 1/L * (1-z^-L / 1-z^-1)

    % Filterkoeffizienten erstellen
    numCoeff      = zeros(1, L+1);
    numCoeff(1)   = 1;
    numCoeff(L+1) = -1;

    denumCoeff    = [1 -1];

    % Filtern
    y = 1/L * filter(numCoeff, denumCoeff, y1bit);

    % Unterabtasten
    y = y(1:L:end);


    % Uebertrgungsfunktion des Dezimationsfilters
    freqz(numCoeff, denumCoeff, 512)
    title('Uebertragungsfunktion des Dezimationsfilters')

end
```