Kommunikationstechnik II – Wintersemester 07/08

Prof. Dr. Stefan Weinzierl

Musterlösung: 3. Aufgabenblatt

1. Aufgabe: Up-/Downsampling

Die Abtastfolge $x[n]$ wird mit dem Faktor M unter- und dem Faktor L überabgetastet.

a) Stellen Sie die beiden Blockdiagramme mit Hilfe eines analytischen Ausdrucks dar. Wie groß ist in beiden Fällen die neue Abtastrate f_S und das Abtastintervall T_S?

Im Fall von Downsampling:

$$x[n] = x(nT_S) \;\rightarrow\; y_d[n] = x(nMT_S)$$

$$f_S = \frac{1}{T_S} \qquad \rightarrow f_S' = \frac{f_S}{M} = \frac{1}{MT_S}$$

$$T_S' = MT_S = \frac{M}{f_S}$$

Im Fall von Upsampling:

$$x[n] = x(nT_S) \;\rightarrow\; y_u[n] = \begin{cases} x(\frac{nT_S}{L}), & n = 0, \pm L, \pm 2L, \ldots \\ 0, & \text{sonst} \end{cases}$$

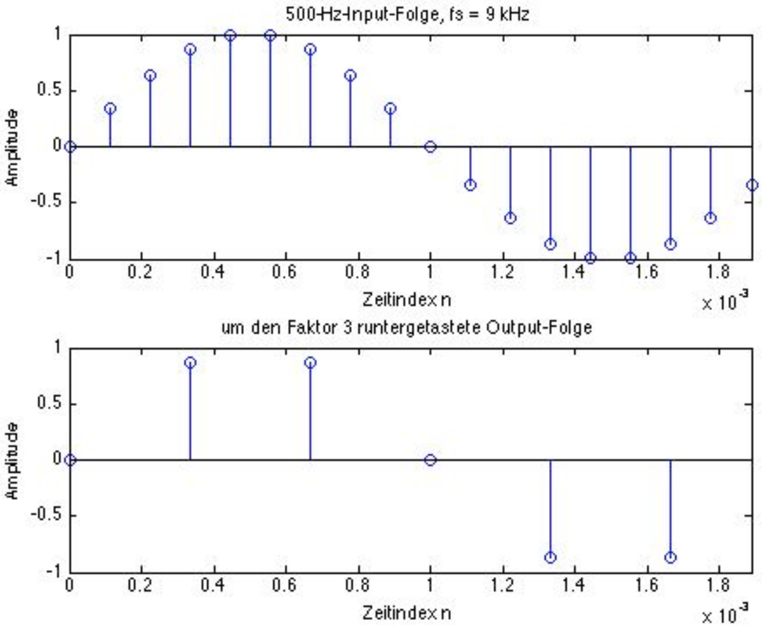
$$f_S = \frac{1}{T_S} \qquad \rightarrow f_S' = Lf_S = \frac{L}{T_S}$$

$$T_S' = \frac{T_S}{L} = \frac{1}{Lf_S}$$

b) Stellen Sie die Über- und Unterabtastung in Matlab dar für ein Signal mit der Frequenz 500 Hz, das ursprünglich mit 9 kHz abgetastet wird. Es sei $M = L = 3$.
Tipp: Füllen Sie bei der Überabtastung im Signalvektor Nullen für die neuen Abtast-werte ein.

```
% ---------Downsampling-----------------%

% Erzeuge Sinusfolge
fs = 9000;
f = 500;
n = 0:1/fs:1-1/fs;
x = sin(2*pi*f*n);
```

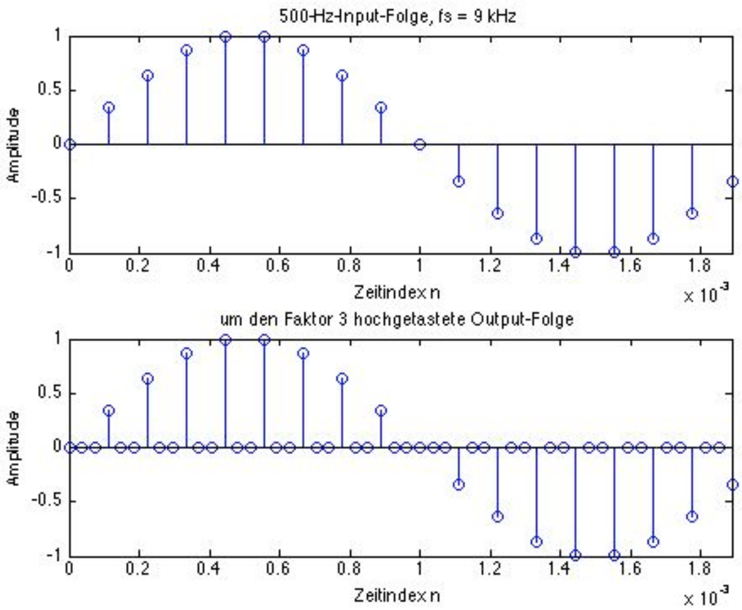
```
% Erzeuge runtergetastete Folge mit zugehörigem Zeitvektor n_M
M = 3;
y_down = x([1 : M : length(x)]);
n_M = 0:M/fs:1-M/fs;

% Plot des Input- und des Outputsignals über eine Periode des Signals
index_x = round(fs/f);
index_ydown = round((fs/M)/f);
figure
subplot(2,1,1)
stem(n(1:index_x), x(1:index_x));
title('500-Hz-Input-Folge, fs = 9 kHz')
xlabel('Zeitindex n'), ylabel('Amplitude')
axis([0 n(index_x) -1 1])
subplot(2,1,2)
stem(n_M(1:index_ydown), y_down(1:index_ydown));
title(['um den Faktor 3 runtergetastete Output-Folge'])
xlabel('Zeitindex n'), ylabel('Amplitude')
axis([0 n(index_x) -1 1])
```



Nach der Unterabtastung liegt das Signal M mal weniger samples pro Periode vor. Um Aliasing zu vermeiden, muss ein Tiefpassfilter vorgeschaltet werden (siehe Aufgabe c)).

```
% ---------Upsampling-----------------%

% Erzeuge hochgetastete Folge mit zugehörigem Zeitvektor n_L
L = 3;
% y_up =interp(x,L);
y_up = zeros(1, L*length(x));
y_up([1: L: length(y_up)]) = x;
n_L = 0:1/(fs*L):1-1/(fs*L);

% Plot des Input- und des Outputsignals ?ber eine Periode des Signals
index_yup = round((fs*L)/f);
figure
subplot(2,1,1)
stem(n(1:index_x), x(1:index_x));
```

```
title('500-Hz-Input-Folge, fs = 9 kHz')
xlabel('Zeitindex n'), ylabel('Amplitude')
axis([0 n(index_x) -1 1])
subplot(2,1,2)
stem(n_L(1:index_yup), y_up(1:index_yup));
title(['um den Faktor 3 hochgetastete Output-Folge'])
xlabel('Zeitindex n'), ylabel('Amplitude')
axis([0 n(index_x) -1 1])
```



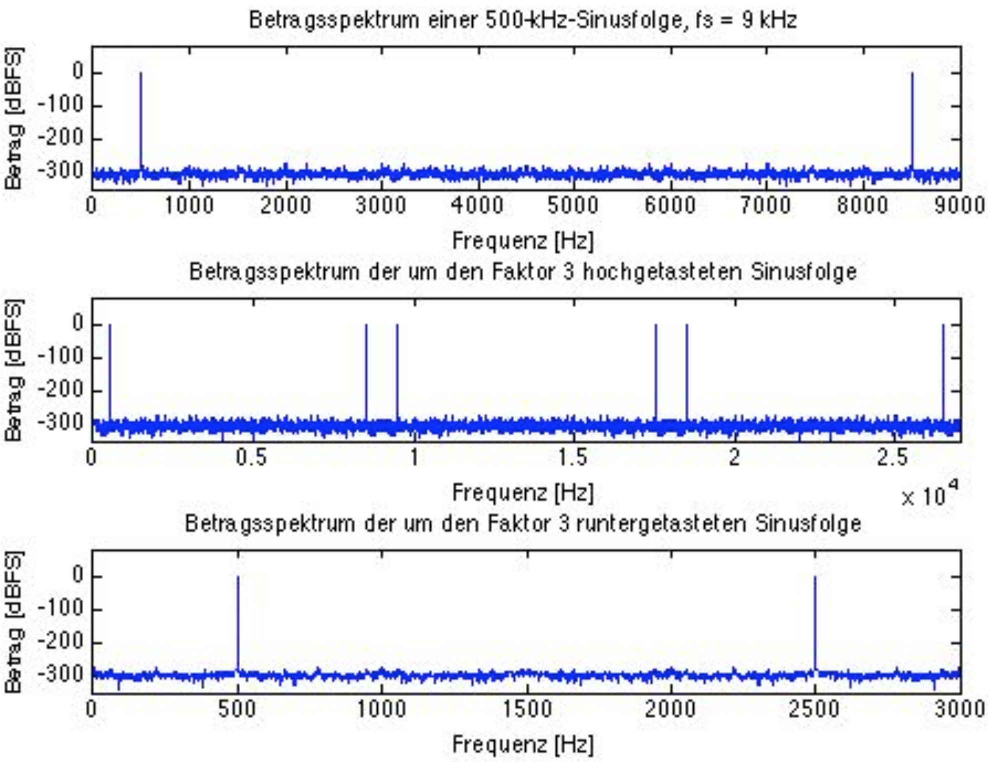
Nach der Überabtastung liegt das Signal L mal mehr samples pro Periode vor. Die zusätzlichen samples werden vorerst mit Nullen aufgefüllt, sie müssen jedoch anschließend durch Tiefpassfilterung zu Abtastwerten interpoliert werden (siehe Aufgabe c)).

c) Stellen Sie das ursprüngliche und die beiden neu abgetasteten Signale im Frequenzbereich dar (in Matlab oder als Skizze). Erklären Sie anhand des Ergebnisses, welche Filter zusätzlich nötig sind.

```
% normierte FFT der Signale
X = abs(fft(x))/max(abs(fft(x)));
Y_down = abs(fft(y_down))/max(abs(fft(y_down)));
Y_up = abs(fft(y_up))/max(abs(fft(y_up)));

% Plots der FFTs
N = length(X);
N_M = length(Y_down);
N_L = length(Y_up);
f_indexx = [0:fs/N:fs-1];
f_indexydown = [0:(fs/M)/N_M:(fs/M)-1];
f_indexyup = [0:(fs*L)/N_L:(fs*L)-1];

figure
subplot (3,1,1), plot(f_indexx,20*log10(X))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum einer 500-kHz-Sinusfolge, fs = 9 kHz')
axis([0 fs -350 80])
```

```
subplot (3,1,2), plot(f_indexyup,20*log10(Y_up))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum der um den Faktor 3 hochgetasteten Sinusfolge')
axis([0 fs*L -350 80])
subplot (3,1,3), plot(f_indexydown,20*log10(Y_down))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum der um den Faktor 3 runtergetasteten Sinusfolge')
axis([0 fs/M -350 80])
```

Upsampling:
Durch die erste Abtastung des Signals mit 9 kHz tritt eine Spiegelfrequenz bei f_S - f auf, also bei 8,5 kHz. Wird dieses Signal nun überabgetastet, so ergeben zusätzliche Spiegelfrequenzen um jedes Vielfache der alten Abtastfrequenz.
Eine anschließende Tiefpassfilterung bei der halben alten Abtastfrequenz bewirkt im Frequenzbereich die Unterdrückung der entstandenen Spiegelfrequenzen zwischen $f_{S\_alt}/2$ und f_{S_neu} - $f_{S\_alt}/2$. Im Zeitbereich erfolgt die Interpolation zwischen den durch die eingefügten Nullen „getrennten" samples. Das Tiefpassfilter wird deshalb Interpolations- oder Anti-Imaging-Filter genannt. Der Vorgang der Überabtastung mit anschließender Tiefpassfilterung wird Interpolation genannt.
Das bei der Rückwandlung in ein analoges Signal notwendige Rekonstruktionsfilter (siehe Aufgabenblatt 2) unterdrückt auch noch die Signalanteile zwischen $f_{S\_alt}/2$ und f_{S_neu}, es muss jedoch dank der Überabtastung nicht besonders steilflankig sein.



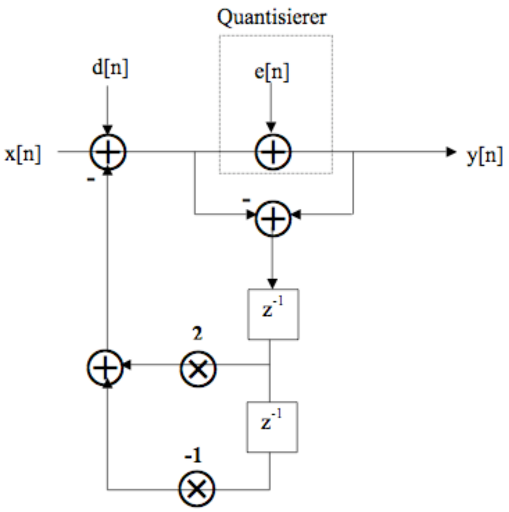
Downsampling:
Beim Downsampling wird die Abtastrate und somit auch die Spiegelfrequenzen bei f_S - f in tiefere Frequenzbereiche verschoben. Um zu vermeiden, dass durch diese Aliasing auftritt, dass also $f_{S\_neu} - f < f$ bzw. $f_{S\_neu} < 2f$, muss das Signal *vor* dem Downsampling mit einem Anti-Aliasing- oder Dezimationsfilter bandbegrenzt werden.

Der Vorgang der Unterabtastung mit vorhergehender Tiefpassfilterung wird Dezimation genannt.

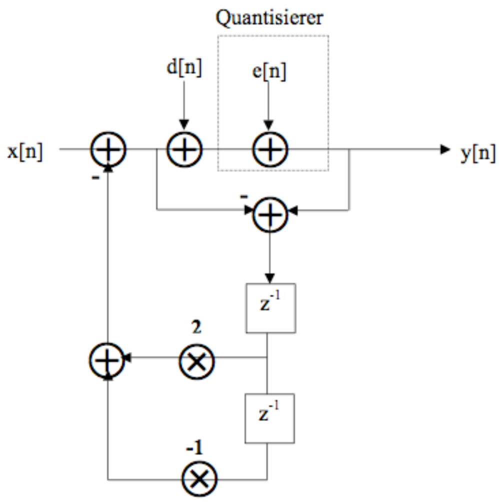
d) Nennen Sie die zwei Hauptgründe für Überabtastung.

Der erste Grund für Überabtastung liegt im Aufwand für den Entwurf des analogen Antialiasing-Filters. Dieses müsste im Normalfall sehr steilflankig sein, damit die gesamte Bandbreite des Nutzsignals bis zur Nyquistfrequenz genutzt werden kann. Wird die Abtastrate aber höher gesetzt, so verschiebt sich auch die Nyquistfrequenz zu einer höheren Frequenz, das Nutzband bleibt jedoch gleich. Es ist also ein flacheres Filter ausreichend, um trotzdem die gesamte Bandbreite des Nutzsignals zu übertragen. Ebenso kann das Rekonstruktionsfilter deutlich flacher ausfallen (siehe Aufgabe c)).
Der zweite Grund bezieht sich auf den Quantisierungsfehler. Dessen Rauschleistung ist – bei guter Aussteuerung und ausreichender Wortbreite – über das gesamte Spektrum gleichverteilt, gleichzeitig ist sie jedoch nicht abhängig von der Abtastfrequenz. Steigt also die Bandbreite des Übertragungssignals mit der Abtastrate, so verteilt sich die Rauschleistung über einen größeren Frequenzbereich. Nach erneuter (digitaler) Tiefpassfilterung resultiert ein größerer SNR, der mit ca. 3 dB pro Frequenzverdoppelung steigt.

2. Aufgabe: Noiseshaping



A



B

a) Stellen Sie die Differenzengleichungen für die beiden dargestellten Noiseshaping-Systeme auf und berechnen Sie jeweils die Signal- und die Rauschübertragungsfunktion.

System A:

$$y[n] = x[n] + d[n] + e[n] - 2e_0[n-1] + e_0[n-2]$$
$$e_0[n] = x[n] + d[n] + e[n] - (x[n] + d[n]) = e[n]$$

somit ist:

$$y[n] = x[n] + d[n] + e[n] - 2e[n-1] + e[n-2]$$

im z-Bereich:

$$Y(z) = X(z) + D(z) + E(z) - 2z^{-1}E(z) + z^{-2}E(z) = X(z) + \underbrace{D(z) + E(z)}_{\text{Rauschen}}\underbrace{\left[1 - 2z^{-1} + z^{-2}\right]}_{\text{spektrale Formung}}$$

Rausch- und Signalübertragungsfunktion:

$$H_X(z) = \frac{Y(z)}{X(z)} = H_D(z) = \frac{Y(z)}{D(z)} = 1$$

$$H_E(z) = \frac{Y(z)}{D(z)} = 1 - 2z^{-1} + z^{-2}$$

System B:

$$y[n] = x[n] + d[n] + e[n] - 2e_0[n-1] + e_0[n-2]$$
$$e_0[n] = x[n] + d[n] + e[n] - x[n] = d[n] + e[n]$$

somit ist:

$$y[n] = x[n] + d[n] + e[n] - 2d[n-1] + d[n-2] - 2e[n-1] + e[n-2]$$

im z-Bereich:

$$Y(z) = X(z) + D(z) + E(z) - 2z^{-1}D(z) + z^{-2}D(z) - 2z^{-1}E(z) + z^{-2}E(z)$$
$$= X(z) + D(z)\left[1 - 2z^{-1} + z^{-2}\right] + E(z)\left[1 - 2z^{-1} + z^{-2}\right] = X(z) + \underbrace{\left[D(z) + E(z)\right]}_{\text{Rauschen}}\underbrace{\left[1 - 2z^{-1} + z^{-2}\right]}_{\text{spektrale Formung}}$$

Rausch- und Signalübertragungsfunktion:

$$H_X(z) = \frac{Y(z)}{X(z)} = 1$$

$$H_E(z) = \frac{Y(z)}{E(z)} = H_D(z) = \frac{Y(z)}{D(z)} = 1 - 2z^{-1} + z^{-2}$$

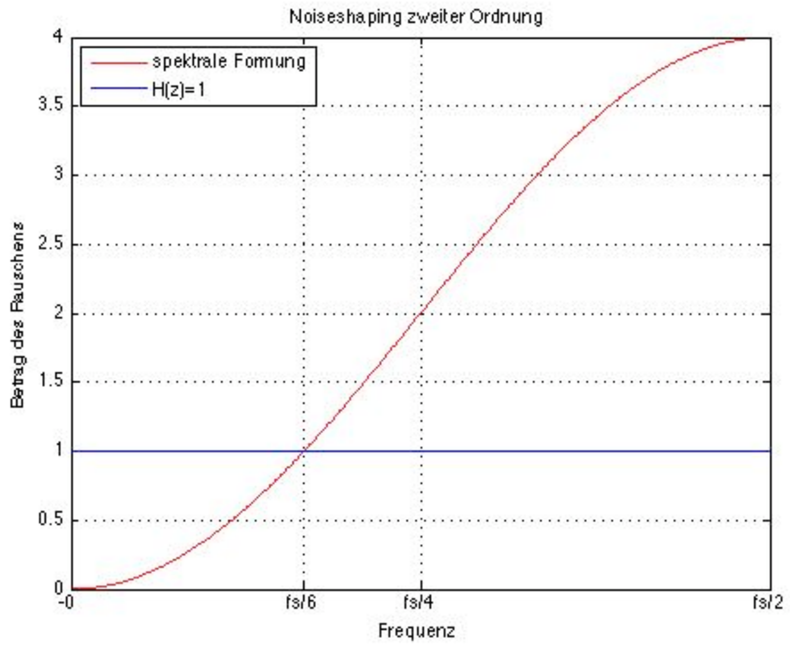
b) Welches System ist sinnvoller? Begründen Sie Ihre Antwort.

Im System A wir zwar der Quantisierungsfehler spektral geformt, nicht aber das Dithersignal. Dieses wird genau wie das Nutzsignal mit *H*(*z*) = 1 übertragen. Im System B hingegen wird auch das Dithersignal bei der Spektralformung berücksichtigt. Dieser Algorithmus wird also bevorzugt.

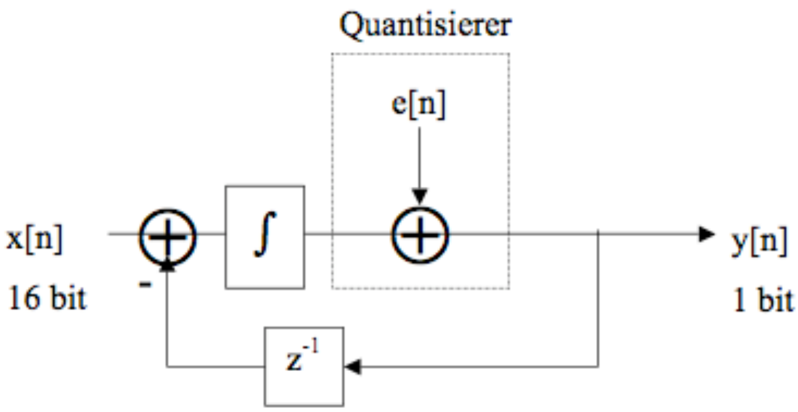
c) Stellen Sie die Rauschübertragungsfunktion des in b) gewählten Systems in Matlab dar.

```
w = 0:2*pi/1024:2*pi;
H = 1-2*exp(-i*w)+exp(-2*i*w);
```

```
plot(w(1:512),abs(H(1:512)),'r')
title('Noiseshaping zweiter Ordnung')
xlabel('Frequenz'), ylabel('Betrag des Rauschens')
axis([0 pi 0 4])
set(gca,'XTick',[0 pi/3 0.5*pi pi])
set(gca,'XTickLabel',{'-0';'fs/6';'fs/4';'fs/2'})
grid on
hold on
plot(w(1:512),ones(1,512))
legend('spektrale Formung','H(z)=1',2)
```



3. Aufgabe: Delta- Sigma-Modulation



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

zurücktransformiert in den Zeitbereich:

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

Dies entspricht einem Noiesshaping erster Ordnung!

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.

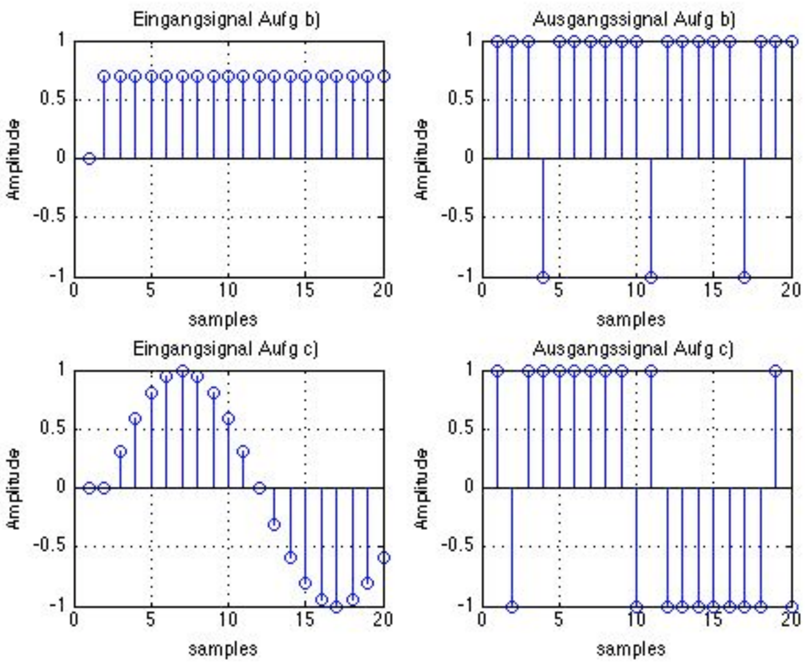
c) Berechnen Sie den Ausgang *y*[*n*] für ein voll ausgesteuertes Sinussignal als Eingang *x*[*n*] mit einer Periode von 20 Samples und stellen Sie Eingangs- und Ausgangssignal über eine Periodendauer dar.

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

```
% 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 ensteht auch der Quantisierungsfehler e(n)
    if zustand >= 0
        quantisierer_ausgang = 1;
    else
        quantisierer_ausgang = -1;
    end

    % Ausgangssignal
    x1_moduliert(n-1) = quantisierer_ausgang;

    %Bestimme den Quantisierungsfehler e(n)
    e(n) = quantisierer_ausgang - 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
        quantisierer_ausgang = 1;
    else
        quantisierer_ausgang = -1;
    end

    x2_moduliert(n-1) = quantisierer_ausgang;
    e(n) = quantisierer_ausgang - 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(x1_moduliert), 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(x2_moduliert), title('Ausgangssignal Aufg c)')
xlabel('samples'), ylabel('Amplitude'), grid on
```



Berechnung der ersten 6 samples aus Aufgabe b):

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

$$y(1) = \left[x(1) - e(0)\right]_Q = \left[0{,}7 - 0{,}3\right]_Q = \left[0{,}4\right]_Q = 1$$
$$e(1) = 1 - 0{,}4 = 0{,}6$$

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

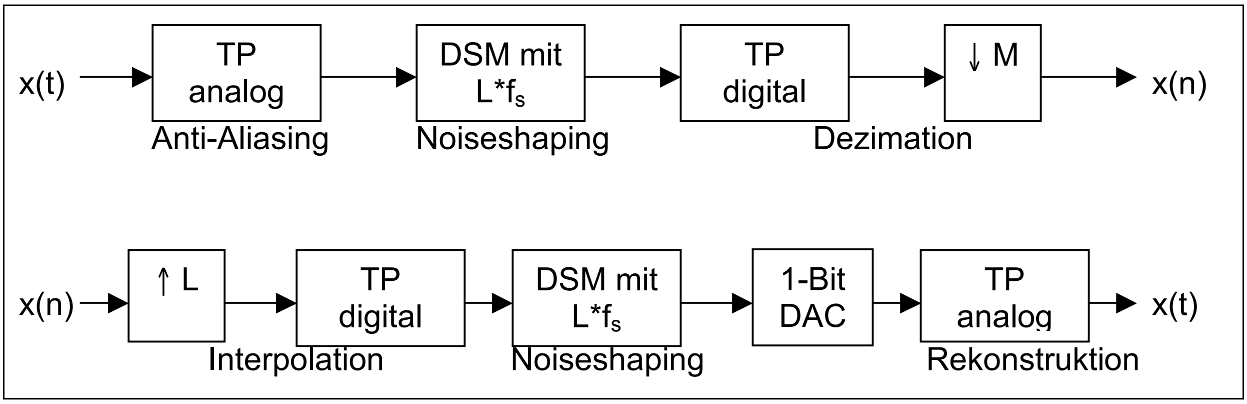
$$y(3) = \left[x(3) - e(2)\right]_Q = \left[0{,}7 - 0{,}9\right]_Q = \left[-0{,}2\right]_Q = -1$$
$$e(3) = -1 + 0{,}2 = -0{,}8$$

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

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

Systematische Einordnung des Delta-Sigma-Modulators:

- ist Grundbaustein von Sigma-Delta-A/D- und -D/A-Wandlern
- es wird die Differenz zwischen überabgetastetem Eingangs- und Ausgangssignal („Delta") integriert („Sigma") und auf sehr niedrige Wortbreite quantisiert
- um einen kompletten A/D- bzw. D/A-Wandler zu realisieren benötigt der δσ-Modulator aber 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 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 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)