Technische Universität Berlin
Prof. Dr. Stefan Weinzierl 28.11.2013

Musterlösung: 25. November 2013, 18:20

1 Abtastung

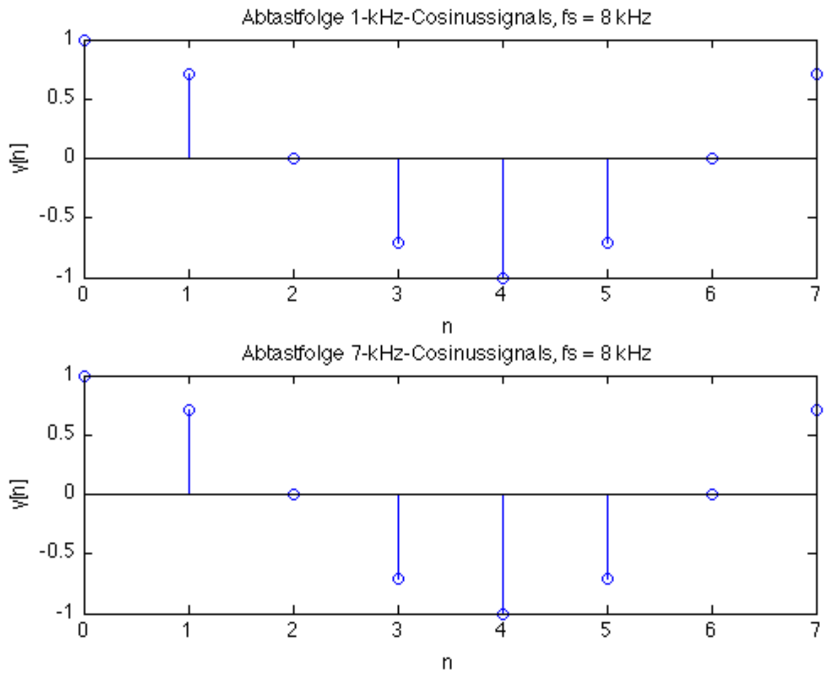
a) Simulieren Sie mit Matlab zwei Cosinussignale der Länge 1 s mit den Frequenzen 1 kHz und 7 kHz. Tasten Sie die beiden Signale mit einer Abtastfrequenz von 8 kHz ab und vergleichen Sie die Abtastfolgen. Wie lässt sich das Ergebnis erklären?

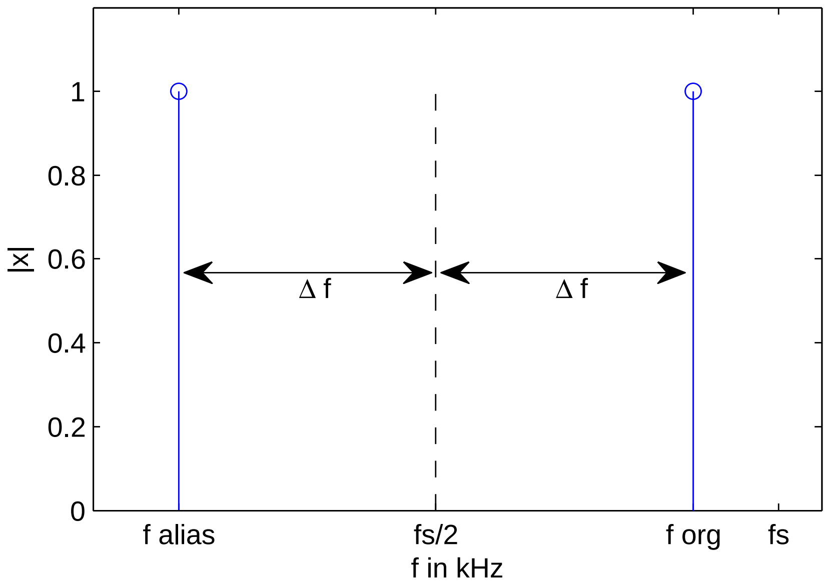
Lösung:

```
%% a)
clear; clc;
% Abtastfrequenz und Wertevektor
fs8 = 8000;
n8 = 0:fs8;

% Abtastung zweier Signale mit Frequenz 1 kHz bzw. 7 kHz
f1 = 1000;
y1 = cos(n8*2*pi*f1/fs8);

f2 = 7000;
y2 = cos(n8*2*pi*f2/fs8);


%% Plot ueber eine Periode des 1-kHz-Signals
index1 = (fs8/f1);
figure(1)
subplot(2,1,1), stem(n8(1:index1), y1(1:index1))
xlabel('n'), ylabel('y[n]')
title('Abtastfolge 1-kHz-Cosinussignals, fs = 8 kHz')
subplot(2,1,2), stem(n8(1:index1), y2(1:index1))
title('Abtastfolge 7-kHz-Cosinussignals, fs = 8 kHz')
xlabel('n'), ylabel('y[n]')

%%
sound(y1,fs8)
pause(1)
sound(y2,fs8)
```

Es zeigt sich, dass die beiden Abtastfolgen nicht zu unterscheiden sind. Dies liegt daran, dass das 7 kHz-Signal mit 8 kHz unterabgetastet wird. Das Abtasttheorem $f_S > 2f_{max}$ wird nicht eingehalten. Die Spiegelfrequenz, welche durch die Abtastung entsteht, liegt exakt bei 8 kHz - 7 kHz = 1 kHz. Die Uneindeutigkeit zeigt sich auch beim Probehören.

$$f_{alias} = f_s/2 - \Delta f$$
$$f_{alias} = f_s/2 - (f - f_s/2)$$
$$f_{alias} = f_s - f$$



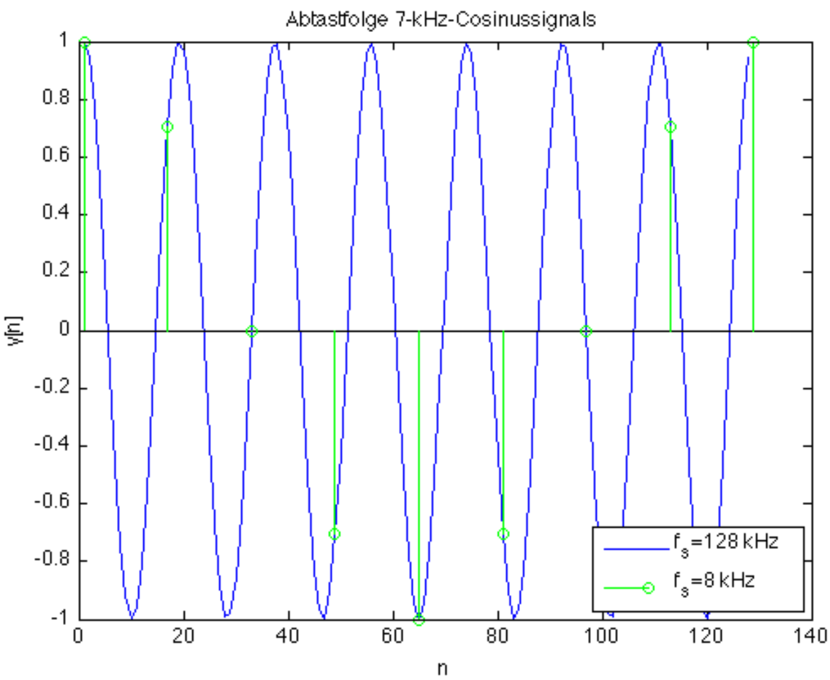


b) Veranschaulichen Sie den bei der Abtastung in Aufgabe a) entstandenen Fehler, indem Sie einen 7 kHz Sinus mit einer Abtastfrequenz von 128 kHz und mit 8 kHz in die selbe Grafik plotten.

Lösung:

```
%% b)

% Abtastfrequenz und Wertevektor
fs128 = 16*fs8;
n128 = 0:fs128;

% Abtastung mit neuer Abtastfrequenz
y128 = cos(n128*2*pi*f2/fs128);
```

```
figure(2)
plot(y128(1:16*index1), 'color', [.7 .7 .7])
% hold on
% stem(1:16:8*16,y2(1:index1))

pause(3)

% y2 in die Grafik von y128 plotten. Weil y128 aber mit dem Faktor 16 Ueberabgetastet
% ist, muessen die Werte von y2 an jedes 16. Sample geplottet werden.
hold on
xlabel('n'), ylabel('y[n]')
title('Abtastfolge 7-kHz-Cosinussignals')
for k = 1:9
    stem((k-1)*16+1, y2(k), 'Color', 'r')
    legend('f_{s}=128 kHz', 'f_{s}=8 kHz', 'location', 'southEast')
    pause(1)
end
hold off
```



Durch die Verletzung des Abtasttheorems wird die 7 kHz Schwingung zu einer 1 kHz Schwingung.

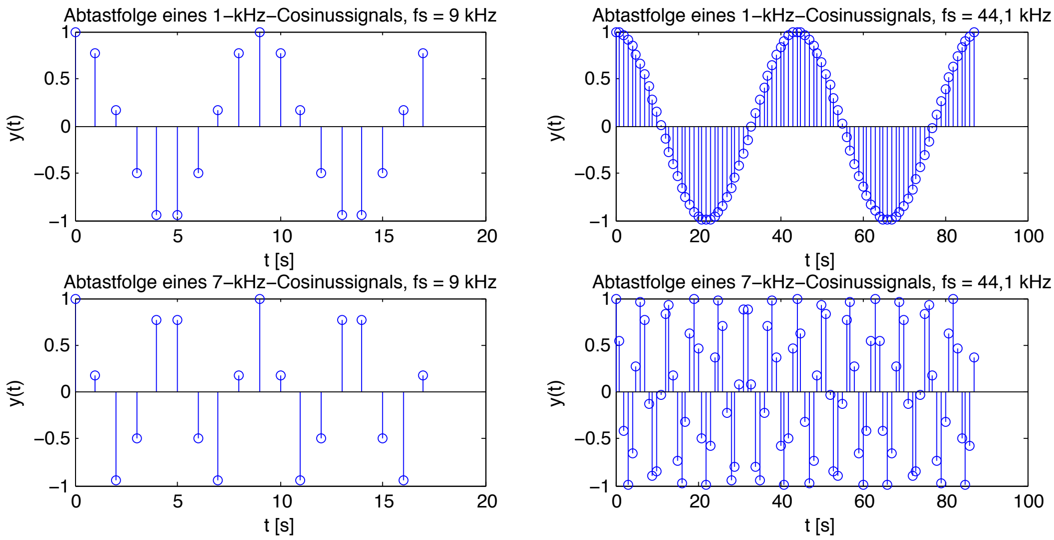
c) Stellen Sie die Betragsspektren der Signale aus a) im Bereich von $0 < f < f_s$ dar. Wie sehen die Betragsspektren für die selben Signalfrequenzen aber mit $f_s = 9\ kHz$ bzw. $f_s = 44,1\ kHz$ aus?

Lösung:

```
%% c)

% neue Abtastfrequenz = 9 kHz und zugeh?riger Zeitvektor
fs9 = 9000;
n9 = 0:fs9;

% Abtastung zweier Signale mit Frequenz 1 kHz bzw. 7 kHz
y3 = cos(n9*2*pi*f1/fs9);
y4 = cos(n9*2*pi*f2/fs9);
```

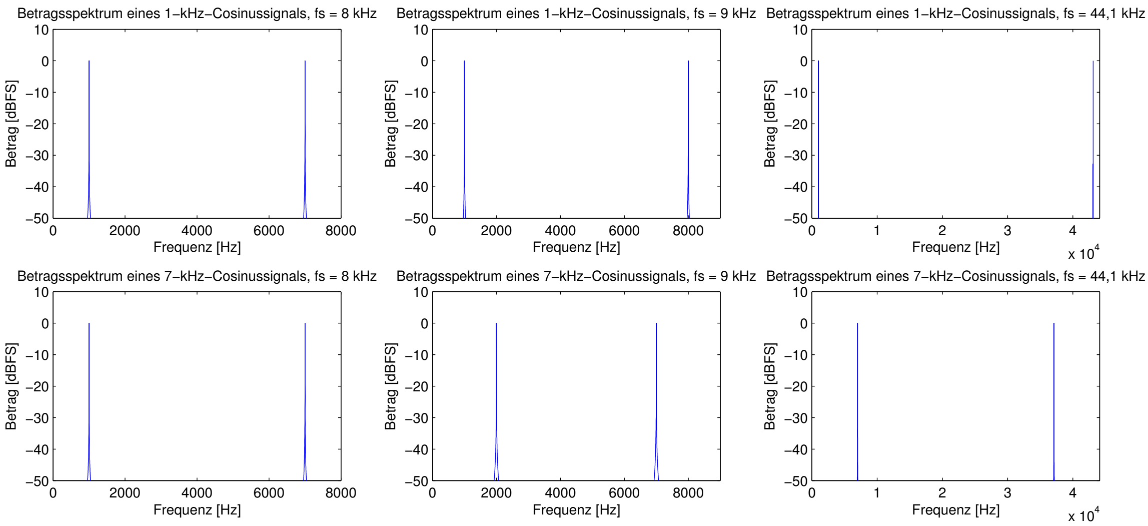
```
% neue Abtastfrequenz = 44,1 kHz und zugeh?riger Zeitvektor
fs44 = 44100;
n44 = 0:fs44;

% Abtastung zweier Signale mit Frequenz 1 kHz bzw. 7 kHz
y5 = cos(n44*2*pi*f1/fs44);
y6 = cos(n44*2*pi*f2/fs44);

% Plot ?ber zwei Perioden des 1-kHz-Signals
index2 = (fs9/f1)*2;
index3 = round((fs44/f1)*2);
figure
subplot(2,2,1), stem(n9(1:index2),y3(1:index2))
xlabel('t [s]'), ylabel('y(t)')
title('Abtastfolge eines 1-kHz-Cosinussignals, fs = 9 kHz')
subplot(2,2,3), stem(n9(1:index2),y4(1:index2))
xlabel('t [s]'), ylabel('y(t)')
title('Abtastfolge eines 7-kHz-Cosinussignals, fs = 9 kHz')
subplot(2,2,2), stem(n44(1:index3),y5(1:index3))
xlabel('t [s]'), ylabel('y(t)')
title('Abtastfolge eines 1-kHz-Cosinussignals, fs = 44,1 kHz')
subplot(2,2,4), stem(n44(1:index3),y6(1:index3))
xlabel('t [s]'), ylabel('y(t)')
title('Abtastfolge eines 7-kHz-Cosinussignals, fs = 44,1 kHz')


%%
sound(y3,fs9)   % 1 kHz mit 9 kHz abgetastet
pause(1)
sound(y4,fs9)   % 7 kHz mit 9 kHz abgetastet
pause(1)
sound(y5,fs44)   % 1 kHz mit 44,1 kHz abgetastet
pause(1)
sound(y6,fs44)   % 7 kHz mit 44,1 kHz abgetastet


%%

% Normierte Darstellung im Frequenzbereich von 0 bis fs
Y1=abs(fft(y1))/max(abs(fft(y1)));
Y2=abs(fft(y2))/max(abs(fft(y2)));
Y3=abs(fft(y3))/max(abs(fft(y3)));
Y4=abs(fft(y4))/max(abs(fft(y4)));
Y5=abs(fft(y5))/max(abs(fft(y5)));
Y6=abs(fft(y6))/max(abs(fft(y6)));
N1 = length(Y1);
N2 = length(Y3);
N3 = length(Y5);
f_index1 = 0:fs8/N1:fs8-fs8/N1;
f_index2 = 0:fs9/N2:fs9-fs9/N2;
f_index3 = 0:fs44/N3:fs44-fs44/N3;

figure
subplot (2,3,1), plot(f_index1,20*log10(abs(Y1)))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum eines 1-kHz-Cosinussignals, fs = 8 kHz')
axis([0 8000 -50 10])
subplot (2,3,4), plot(f_index1,20*log10(abs(Y2)))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
```

```
title('Betragsspektrum eines 7-kHz-Cosinussignals, fs = 8 kHz')
axis([0 8000 -50 10])
subplot (2,3,2), plot(f_index2,20*log10(abs(Y3)))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum eines 1-kHz-Cosinussignals, fs = 9 kHz')
axis([0 9000 -50 10])
subplot (2,3,5), plot(f_index2,20*log10(abs(Y4)))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum eines 7-kHz-Cosinussignals, fs = 9 kHz')
axis([0 9000 -50 10])
subplot (2,3,3), plot(f_index3,20*log10(abs(Y5)))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum eines 1-kHz-Cosinussignals, fs = 44,1 kHz')
axis([0 44100 -50 10])
subplot (2,3,6), plot(f_index3,20*log10(abs(Y6)))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum eines 7-kHz-Cosinussignals, fs = 44,1 kHz')
axis([0 44100 -50 10])
```



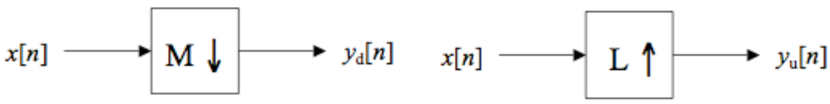
Die beiden Abtastfolgen unterscheiden sich zwar bei einer Abtastung mit 9 kHz, das 7 kHz-Signal ist aber immer noch unterabgetastet. Erst bei einer Abtastrate größer als die doppelte Signalfrequenz (in diesem Fall $44,1$ kHz) können beide Signale eindeutig rekonstruiert werden (Abtasttheorem). Eine Hörkontrolle macht dies deutlich.
Die Betragspektren zeigen, was bereits im Zeitbereich bzw. beim Hören deutlich wurde: Die Spektren aus Aufgabe a) unterscheiden sich nicht. Werden die Signale mit 9 kHz abgetastet, so ergeben sich zwar unterschiedliche Spektren, beim 7-kHz-Signal erscheint aber trotzdem eine Spiegelfrequenz unter der Signalfrequenz (9 kHz - 7 kHz = 2 kHz), so-dass diese als Grundton aufgefasst wird. Um dieses Aliasing zu vermeiden, muss vor der Abtastung ein Tiefpassfilter eingesetzt werden (Anti-Aliasing-Filter), das Signalanteile oberhalb $f_s/2$ (Nyquistfrequenz) unterdrückt. Bei einer Abtastung mit 44,1 kHz ergeben sich zwar auch Spiegelfrequenzen bei $f_s - f$, allerdings erscheinen diese oberhalb der höchsten Nutzfrequenz, und sie sind nicht mehr hörbar. Für die in diesem Fall durch das Hörvermögen hervorgerufene Filterung sorgt im Allgemeinen ein Rekonstruktionstiefpassfilter mit einer Grenzfrequenz bei $f_s/2$. Es zeigt sich, dass die Rauschpegel der im Verhältnis zur Nutzfrequenz höher abgetasteten Signale leicht tiefer sind. Dies liegt an der Verteilung der Fehlerenergie über das gesamte Spektrum (siehe Oversampling).

Matlab-Funktionen: stem, fft



2 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' ?

Lösung: Downsampling:

$$y_d[n] = x(nM) \quad , n = 0,\ 1,\ 2,\ \ldots$$

$$\begin{aligned} f_s = \frac{1}{T_s} \quad \Rightarrow f_s' &= \tfrac{f_s}{M} = \tfrac{1}{MT_s} \\ T_s' &= MT_s = \tfrac{M}{f_s} \end{aligned}$$

Upsampling:

$$y_u[n] = \begin{cases} x\left(\frac{n}{L}\right) & , n = 0,\ L,\ 2L,\ \ldots \\ 0 & , sonst \end{cases}$$

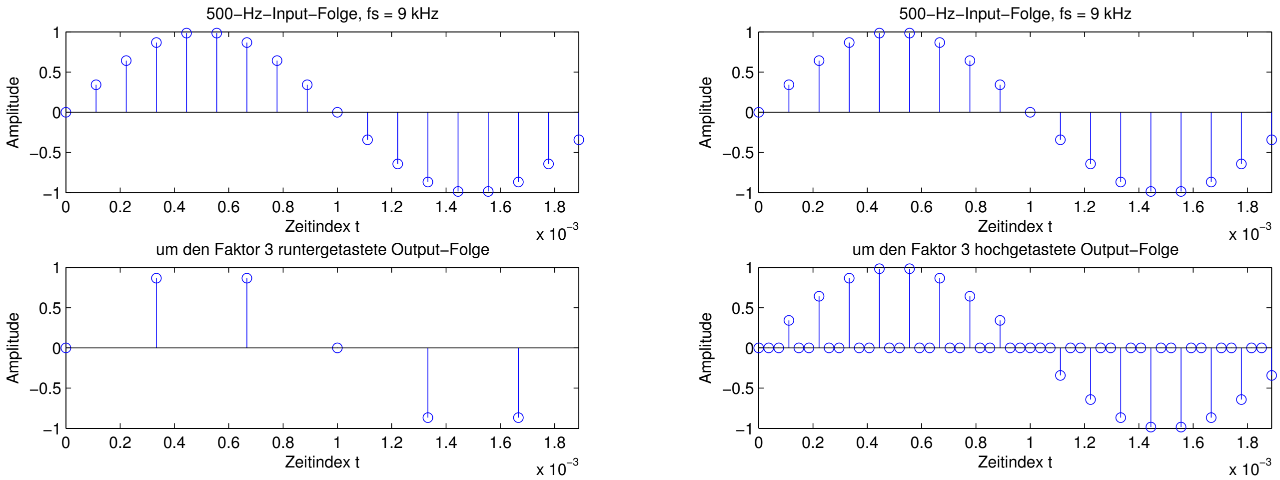
$$\begin{aligned} f_s = \frac{1}{T_s} \quad \Rightarrow f_s' &= Lf_s = \tfrac{L}{T_s} \\ T_s' &= \tfrac{T_s}{L} = \tfrac{1}{Lf_s} \end{aligned}$$

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 $M = L = 3$.

Lösung:

```
%% b)

% Erzeuge Sinusfolge
```

```
fs = 9000;
f = 500;
t = 0:1/fs:1-1/fs;
x = sin(2*pi*f*t);

% Erzeuge runtergetastete Folge mit zugehoerigem Zeitvektor t_M
M = 3;
y_down = x(1 : M : length(x));
t_M = 0:M/fs:1-M/fs;

% Plot des Input- und des Outputsignals ueber eine Periode des Signals
index_x = round(fs/f);
index_ydown = round((fs/M)/f);
figure(2)
subplot(2,1,1)
stem(t(1:index_x), x(1:index_x));
title('500-Hz-Input-Folge, fs = 9 kHz')
xlabel('Zeitindex t'), ylabel('Amplitude')
axis([0 t(index_x) -1 1])
subplot(2,1,2)
stem(t_M(1:index_ydown), y_down(1:index_ydown));
title('um den Faktor 3 runtergetastete Output-Folge')
xlabel('Zeitindex t'), ylabel('Amplitude')
axis([0 t(index_x) -1 1])


%% -- Upsampling --%

% Erzeuge hochgetastete Folge mit zugehoerigem Zeitvektor t_L
L = 3;
y_up = zeros(1, L*length(x));
y_up(1: L: length(y_up)) = x;
t_L = 0:1/(fs*L):1-1/(fs*L);

% Plot des Input- und des Outputsignals ueber eine Periode des Signals
index_yup = round((fs*L)/f);
figure(3)
subplot(2,1,1)
stem(t(1:index_x), x(1:index_x));
title('500-Hz-Input-Folge, fs = 9 kHz')
xlabel('Zeitindex t'), ylabel('Amplitude')
axis([0 t(index_x) -1 1])
subplot(2,1,2)
stem(t_L(1:index_yup), y_up(1:index_yup));
title('um den Faktor 3 hochgetastete Output-Folge')
xlabel('Zeitindex t'), ylabel('Amplitude')
axis([0 t(index_x) -1 1])
```

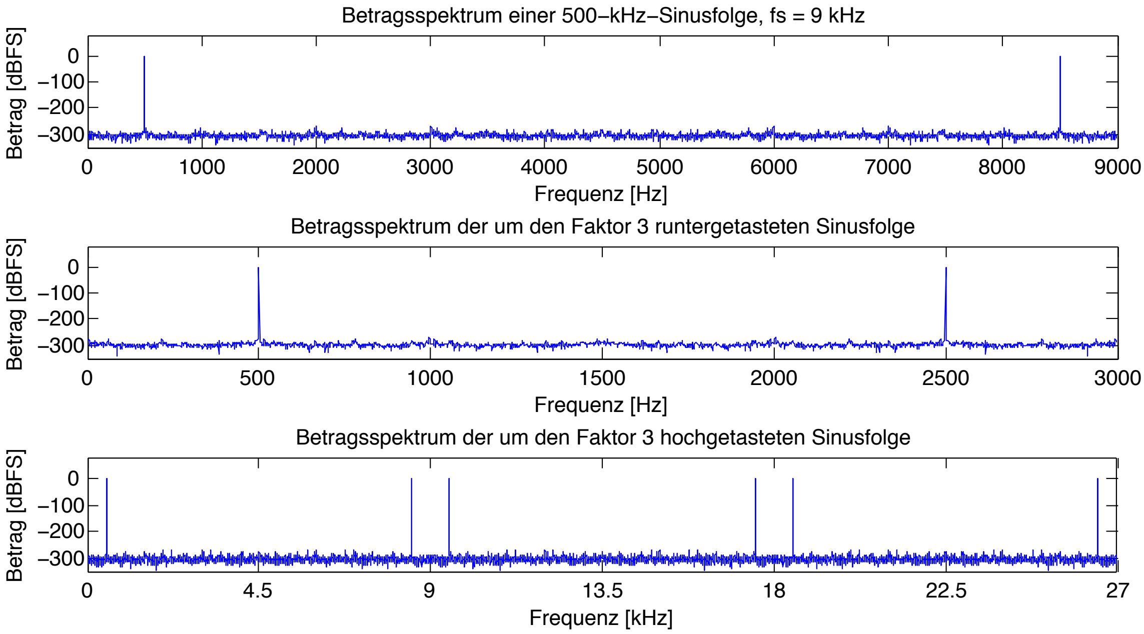


Nach der Unterabtastung liegt das Signal M mal weniger samples pro Periode vor. Um Aliasing zu vermeiden, muss ein Tiefpassfilter (Anti-Aliasing- oder Dezimations-Filter) vorgeschaltet werden. Nach der Überabtastung hat das Signal L mal mehr samples. Die zusätzlichen samples werden vorerst mit Nullen aufgefüllt, sie müssen jedoch anschließend durch Tiefpassfilterung (Anti-Imaging- oder Interpolationsfilter) zu Abtastwerten interpoliert werden.

c) Stellen Sie das ursprüngliche und die beiden neu abgetasteten Signale im Frequenzbereich $0 \leq f \leq f_s/2$ dar (in Matlab oder als Skizze). Erklären Sie anhand des Ergebnisses, welche Filter zusätzlich nötig sind.

Lösung:

```
%% c)

% 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-fs/N;
f_indexydown = 0:(fs/M)/N_M:(fs/M)-(fs/M)/N_M;
f_indexyup = 0:(fs*L)/N_L:(fs*L)-(fs*L)/N_L;

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_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])
subplot (3,1,3), plot(f_indexyup/1000,20*log10(Y_up))
xlabel('Frequenz [kHz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum der um den Faktor 3 hochgetasteten Sinusfolge')
set(gca, 'xTick', 0:fs/2:L*fs)
axis([0 fs*L/1000 -350 80])
```



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.

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 unterdrückt auch noch die Signalanteile zwischen $f_{s,\,alt/2}$ und $f_{s,\,neu}$, es muss jedoch dank der Überabtastung nicht besonders steilflankig sein.

d) Nennen Sie die zwei Hauptgründe für Überabtastung.
Lösung: 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.
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.