

Musterlösung: 1. November 2012, 15:56

1 Autokorrelation und Leistungsdichtespektrum von Rauschen

Die Autokorrelationsfunktion eines Signals ist definiert als der Erwartungswert des Produktes zweier Amplitudenwerte desselben Zufallssignals zu unterschiedlichen Zeitpunkten und hängt bei stationären Signalen nur von der Verschiebung dieser Zeitpunkte zueinander ab:

$$\varphi_{xx}(\tau) = E\{X_t \cdot X_{t-\tau}\}$$

Ihre Fouriertransformierte, das Leistungsdichtespektrum, liefert Informationen über die spektrale Verteilung der Leistung eines Signals. Gegeben sei ein bandbegrenztes, weißes Rauschsignal mit dem Leistungsdichtespektrum

$$S_{xx}(\omega) = \begin{cases} S_0 & , |\omega| \leq \omega_0 \\ 0 & , \text{sonst} \end{cases}$$

a) Berechnen Sie die Leistung des Rauschsignals

Lösung: Die Leistung kann über das Leistungsdichtespektrum berechnet werden

$$P = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{xx}(\omega) d\omega$$

Hier ist dann:

$$P = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} S_0 d\omega = \frac{S_0}{2\pi} \omega \Big|_{-\omega_0}^{\omega_0} = \frac{2\omega_0 S_0}{2\pi} = 2 \frac{2\pi f_0 S_0}{2\pi} = 2S_0 f_0$$

b) Berechnen Sie die Autokorrelation $\varphi_{xx}(\tau)$ des Signals

Lösung: Die Autokorrelation ist die inverse Fouriertransformierte des Leistungsdichtespektrums

$$\varphi_{xx}(\tau) = F^{-1}\{S_{xx}(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{xx}(\omega) \cdot e^{j\omega\tau} d\omega$$

In unserem Fall ist also:

$$\begin{aligned} \varphi_{xx}(\tau) &= \frac{S_0}{2\pi} \int_{-\omega_0}^{\omega_0} e^{j\omega\tau} d\omega = \frac{S_0}{2\pi} \left(\int_{-\omega_0}^{\omega_0} \cos(\omega\tau) d\omega + j \underbrace{\int_{-\omega_0}^{\omega_0} \sin(\omega\tau) d\omega}_{=0} \right) \\ &= \frac{S_0}{2\pi\tau} \sin(\omega\tau) \Big|_{-\omega_0}^{\omega_0} = S_0 \frac{\sin(\omega_0\tau) - \sin(-\omega_0\tau)}{2\pi\tau} \end{aligned}$$

wegen der Symmetrie des Sinus ($\sin(x) = -\sin(-x)$) kann vereinfacht werden:

$$S_0 \frac{2 \sin(\omega_0 \tau)}{2\pi\tau} = S_0 \frac{f_0}{f_0} \cdot \frac{2 \sin(2\pi f_0 \tau)}{2\pi\tau} = 2S_0 f_0 \cdot \frac{\sin(2\pi f_0 \tau)}{2\pi f_0 \tau} = 2S_0 f_0 \text{si}(\omega_0 \tau)$$

c) Bestimmen Sie aus der AKF die Leistung des Signals und vergleichen Sie das Ergebnis mit dem Wert aus a).

Lösung: Die Leistung ist gleich der AKF an der Stelle $\tau = 0$.

$$P = \varphi_{xx}(\tau = 0) = 2S_0 f_0$$

Beide Berechnungsmethoden führen auf ein identisches Ergebnis.

d) Geben Sie die AKF desselben Signals an, diesmal für einen nicht Bandbegrenzten Fall. Wie interpretieren Sie das Ergebnis im Vergleich zu Aufgabe b)?

Lösung: Geht ω_0 gegen Unendlich, geht die Si-Funktion in einen Deltaimpuls über:

$$\varphi_{xx}(\tau) = \lim_{f_0 \rightarrow \infty} 2S_0 f_0 \text{si}(\omega_0 \tau) = 2S_0 f_0 \delta(\tau)$$

Die Leistung des nicht Bandbegrenzten Rauschens kann der AKF an der Stelle $\tau = 0$ entnommen werden und ist unendlich groß. Die si-Funktion aus Aufgabe b) ist im Vergleich dazu eine Annäherung an den Delta-Impuls, wie sie in der Realität vorkommt.

2 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);

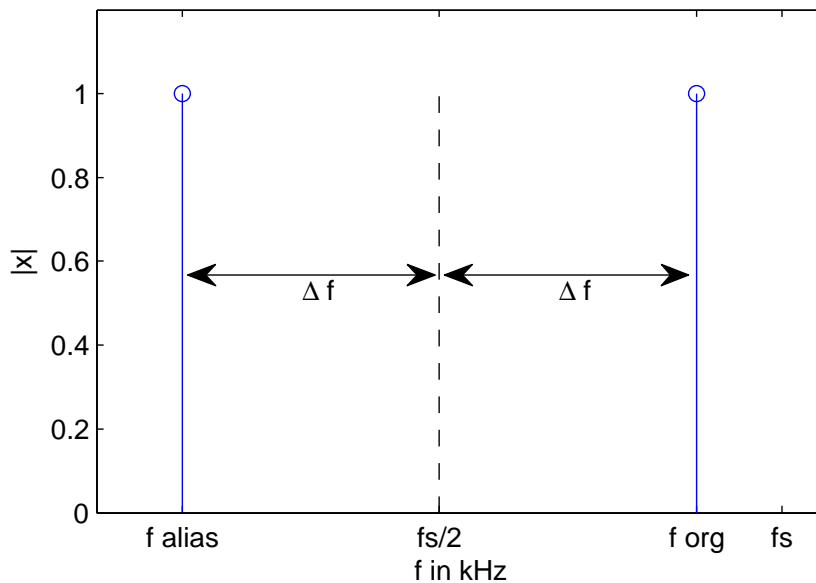
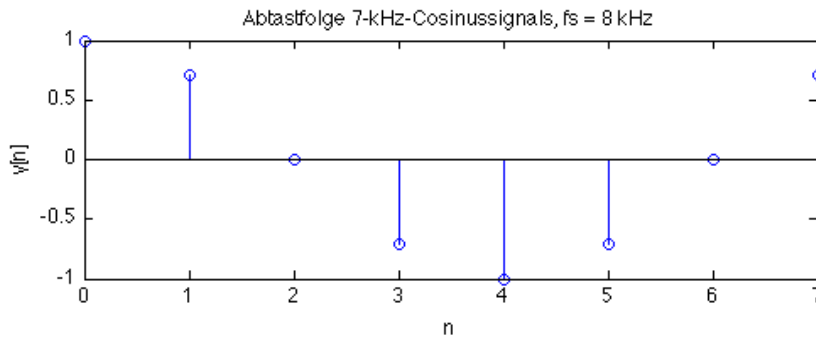
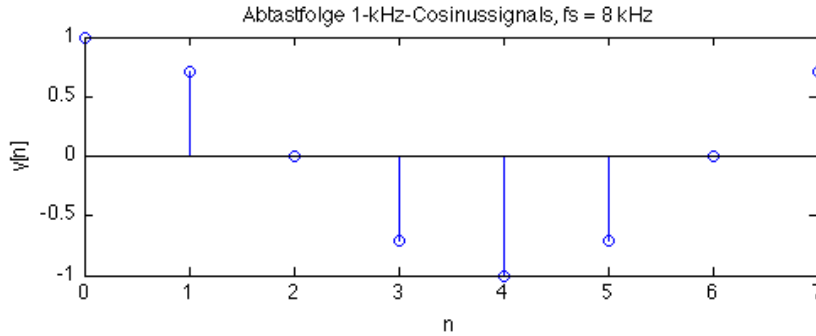
%% Plot ueber eine Periode des 1-kHz-Signals
index1 = (fs8/f1);
figure(1)
subplot(2,1,1), stem(n8(1:index1), y1(1:index1))
xlabel('n'), ylabel('y[n]')
title('Abtastfolge 1-kHz-Cosinussignals, fs = 8 kHz')
subplot(2,1,2), stem(n8(1:index1), y2(1:index1))
title('Abtastfolge 7-kHz-Cosinussignals, fs = 8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

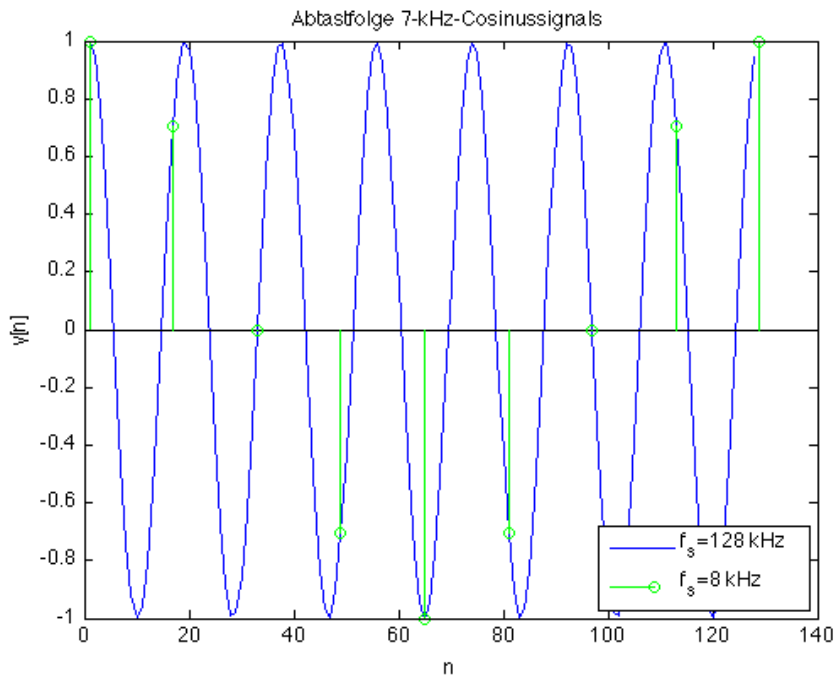
% Abtastung mit neuer Abtastfrequenz
y128 = cos(n128*2*pi*f2/fs128);

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 $f_s = 9 \text{ kHz}$ bzw. $f_s = 44,1 \text{ kHz}$ aus?

Lösung:

```
%% c)

% neue Abtastfrequenz = 9 kHz und zugehöriger 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 zugehöriger 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 über 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

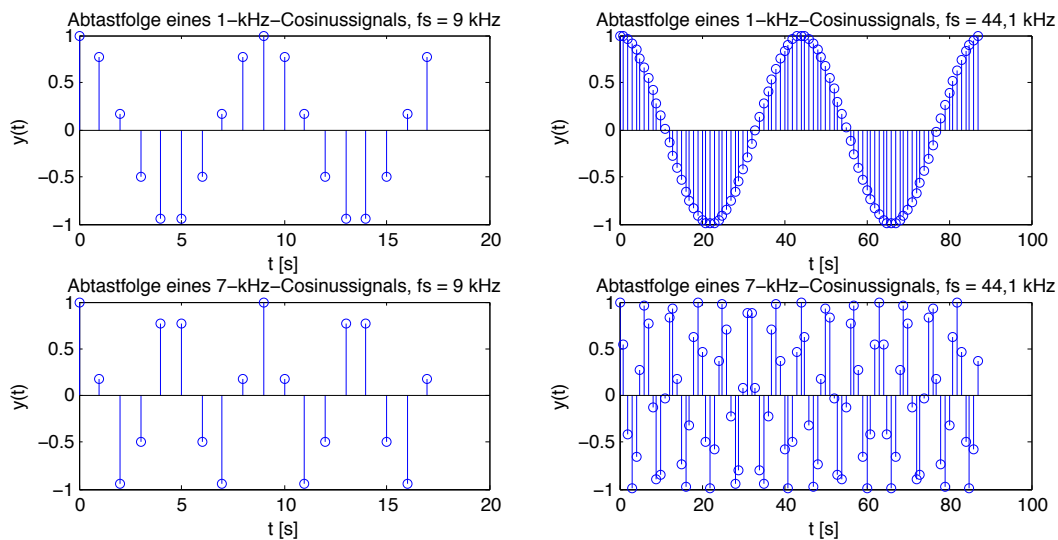
%%

% 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;

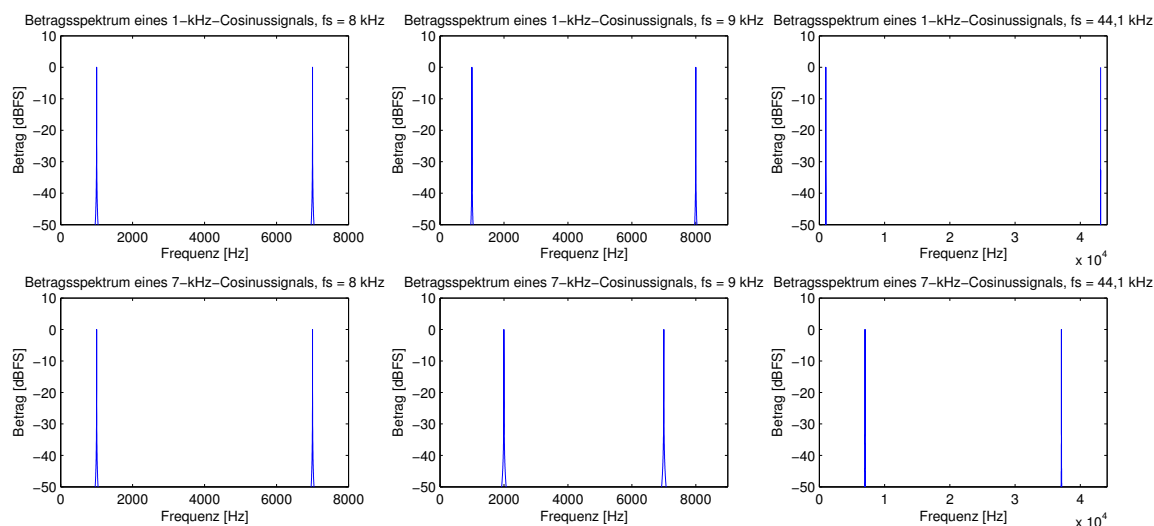
figure
subplot(2,3,1), plot(f_index1,20*log10(abs(Y1)))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum eines 1-kHz-Cosinussignals, fs = 8 kHz')
axis([0 8000 -50 10])
subplot(2,3,4), plot(f_index1,20*log10(abs(Y2)))
xlabel('Frequenz [Hz]'), ylabel('Betrag [dBFS]')
title('Betragsspektrum 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 unterabtastet. Erst bei einer Abtastrate größer als die doppelte Signalfrequenz



(in diesem Fall $44,1 \text{ kHz}$) 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 ($9 \text{ kHz} - 7 \text{ kHz} = 2 \text{ kHz}$), 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 $44,1 \text{ kHz}$ 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`