Spektralanalyse

May 4, 2016

1 Spektralanalyse

Die Betrachtung von Zeitreihen im Frequenzraum ermöglicht die Bestimmung von periodischen Zyklen, z.B. dem Gezeitensignal. Die Zerlegung der Zeitreihe in ihre Frequenzanteile wird auch Harmonische Analyse genannt.

1.1 Fourier-Reihen

Gegeben sei eine periodische Funktion mit der Grundperiode T bzw. Grundfrequenz $f_0 = \frac{1}{T}$

$$x(t) = x(t \pm nT)$$

mit $n = 1, 2, 3, ...$

Mit einigen Ausnahmen lassen sich solche periodischen Daten in Fourier-Reihen entwickeln

$$x(t) = \frac{a_o}{2} + \sum_{n=1}^{\infty}(a_n \cos(2\pi n f_0 t) + b_n \sin(2\pi n f_0 t))$$

Die Fourier-Koeffizienten a_n, b_n erhält man durch Integration über ein Periode T, z.B. $-\frac{T}{2}$ bis $\frac{T}{2}$ oder von 0 bis T

$$a_n = \frac{2}{T} + \int_0^T x(t) \cos(2\pi n f_0 t) dt b_n = \frac{2}{T} + \int_0^T x(t) \sin(2\pi n f_0 t) dt$$

Zu beachten ist, dass $\frac{a_o}{2} = \int_0^T x(t)dt$ der Mittelwert μ_x von $x(t)$ ist. Die Gleichungen können auch in anderer Form, z.B. mit der Kreisfrequenz $\omega = 2\pi f$ und $d\omega = 2\pi df$ geschrieben werden.

1.1.1 Amplituden- und Phasendarstellung

Durch trigonometrische Umformung lässt sich die Fourier-Reihe auch formulieren als

$$x(t) = \frac{a_o}{2} + \sum_{n=1}^{\infty}(\underbrace{A_n}_{=\sqrt{a_n^2+b_n^2}} \cos(2\pi n f_0 t - \underbrace{\Phi_n}_{=\tan^{-1}(\frac{b_n}{a_n})})$$

1.1.2 Darstellung mit komplexen Zahlen

Mittels der Euler-Beziehung $e^{-\imath\Theta} = \cos\Theta - \imath \sin\Theta$ folgt die Darstellung

$$x(t) = \frac{a_o}{2} + \sum_{n=-\infty}^{\infty} A_n e^{\imath \pi n f_0 t}$$

mit $A_n = \frac{1}{2}(a_k - \imath b_k)$ und

$$A_n = \frac{1}{T}\int_0^T x(t)e^{\imath\pi n f_0 t}$$

wobei $n = \pm 1, \pm 2, \pm 3, ...$ nun auch negative Werte annimmt.

Auch wenn $x(t)$ eine reellwertige Zeitreihe ist, kann sie durch komplexwertige negative und positive Frequenzkomponenten beschrieben werden. Dabei gilt der Zusammenhang

$$A_n = |A_n|e^{-\imath\Theta_n}$$

mit $|A_n| = \frac{1}{2}\sqrt{a_n^2 + b_n^2} = \frac{x_n}{2}$ und $\Theta_n = \tan^{-1}(\frac{b_n}{a_n})$.

Für reelle $x(t)$ gelten die Symmetrien

$$|A_{-n}| = |A_n|$$

$$\Theta_{-n} = -\Theta_n$$

$$A_{-n} = |A_{-n}|e^{-\imath\Theta_{-n}} = |A_n|e^{\imath\Theta_n} = A_n^*$$

1.2 Fourier-Transformation

Gegeben sei eine Zeitreihe $x(t)$. Die Fourier-Reihe lässt sich erweitern zu dem Fourier-Integral

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-\imath 2\pi f t}dt$$

$X(f)$ wird auch direkte Fourier-Transformierte oder (Amplituden-)Spektrum genannt und existiert für die Bedingung

$$\int_{-\infty}^{\infty} |x(t)|dt < \infty$$

.

Die inverse Fourier-Transformation

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{\imath 2\pi f t}df$$

ist i.A. eine komplexwertige Funktion

$$X(f) = X_R(f) - \imath X_I(f)$$

mit Real- und Imaginärteil X_R und X_I

$$X_R(f) = |X(f)|\cos(\Theta(f)) = \int_{-\infty}^{\infty} x(t)\cos(2\pi f t)dt$$

$$X_I(f) = |X(f)|\sin(\Theta(f)) = \int_{-\infty}^{\infty} x(t)\sin(2\pi f t)dt$$

bzw.

$$X(f) = \underbrace{|X(f)|}_{Magnitudenspektrum} e^{-\imath \overbrace{\Theta(f)}^{Phasenspektrum}}$$

Das quadrierte Magnitudenspektrum wird auch Leistungsspektrum genannt. Die graphische Darstellung, meist in logarithmischer Form, wird als Periodogramm bezeichnet.

1.3 Diskrete Fourier-Transformation

Für eine stationäre Zeitreihe von theoretisch unendlicher Länge existiert die Fourier-Transformation nicht, denn es gilt

$$\int_{-\infty}^{\infty} |x(t)| dt = \infty$$

Allerdings liegen tatsächlich gemessene Zeitreihen nur über ein endliches Zeitintervall T vor, und die bestimmte Fourier-Transformation

$$X_T(f) = X(f,T) = \int_0^T x(t) e^{-\imath 2\pi f t} dt$$

existiert immer. Für diskrete Frequenzen $f_n = \frac{n}{T}$ mit $n = \pm 1, \pm 2, \pm 3, ...$ ergibt sich die diskrete Fouriertransformation zu

$$X(f_n, T) = T A_n$$

mit

$$A_n = \frac{1}{T} \int_0^T x(t) e^{-\imath 2\pi f_n t} dt$$

Die Fourier-Transformation für diskrete Frequenzen f ist tatsächlich eine Fourier-Reihe.

1.3.1 Nyquist-Frequenz

Wird die Zeitreihe $x(t)$ an N-Punkten im Abstand Δt gemessen (abgetastet), so beträgt die längste Periode

$$T = N \Delta t$$

Dies führt zu der Grundfrequenz $f_0 = \frac{1}{T}$ und der Nyquist-Frequenz (Grenzfrequenz)

$$f_{Nyquist} = \frac{1}{2\Delta t}$$

Bei der Berechnung wird die Zeitreihe so behandelt, als wäre es eine zyklische Zeitreihe mit der Periode T.

1.4 Rechenregeln für die Fouriertransformation

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-\imath 2\pi f t} dt$$

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{\imath 2\pi f t} df$$

Eigenschaft
Zeit-Raum
Frequenz-Raum
Linearität
$ax(t) - by(t)$
$aX(f) - bY(f)$
Zeitversatz

$$x(t - t_0)$$

$X(f)e^{2\pi\imath f t_0}$
Frequenzversatz

$$x(t)e^{2\pi\imath f_0 t}$$

$X(f - f_0)$
Differentiation

$$\frac{dx(t)}{dt}$$

$$2\pi\imath f X(f)$$

n-mal Differentiation

$$\frac{d(x(t))^n}{dt^n}$$

$$(2\pi\imath f)^n X(f)$$

1.5 Fouriertransformations-Paare

$x(t)$
$X(f)$
1
$\delta(f)$
$\delta(t)$
1
$e^{2\pi\imath f_0 t}$
$\delta(f - f_0)$
$x(t - t_0)$
$X(f)e^{-2\pi\imath f t_0}$
$\int_{-\infty}^{\infty} x_1(u)x_2(t - u)du$
$X_1(f)X_2(f)$
Die Delta-Funktion (Distribution) ist $\delta(0) = \infty$ und sonst null. Es gilt

$$\int_{-\infty}^{\infty} \delta(t)dt = 1$$

und

$$\int_{-\infty}^{\infty} x(t)\delta(t - t_0)dt = x(t_0)$$

1.5.1 Beispiel: Lösung einer DGL im Frequenzraum

Die bekannte Differentialgleichung für ein eindimensionales mechanisches System

$$m\frac{d^2y(t)}{dt^2} + c\frac{dy(t)}{dt} + ky(t) = F(t)$$

lässt sich für die anregende Kraft $F(t) = \delta t$ im Frequenzraum einfach lösen. Durch Fouriertransformation auf beiden Seiten erhält man

$$[-(2\pi)^2 m + \imath 2\pi f c + k]Y(f) = 1$$

$$Y(f) = \frac{1}{k - (2\pi)^2 m + \imath 2\pi f c}$$

2 Periodogramm

Gegeben sei eine Zeitreihe $x(t)$. Das Fourier-Integral

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-\imath 2\pi f t} dt$$

wird auch Amplitudenspektrum genannt. Das quadratische Amplitudenspektrum

$$P_{xx}(f) = |X(f)|^2$$

wird auch das Leistungsspektrum genannt. Die graphische Darstellung des Leistungsspektrums wird als Periodogramm bezeichnet.

Für zeitlich diskrete Zeitreihen sind Fourier-Integral und Fourier-Reihe identisch. Die Berechnung der Fourier-Reihe geschieht üblicherweise durch einen schnellen Algorithmus, der FFT (Fast Fourier Transform).

3 Abtasttheorem und Aliasing

Gegeben sei ein Signal welches Informationen mit einer maximalen Frequenz f_{max} enthält. Dieses Signal soll zeitdiskret gemessen (Sampling) und aus den diskreten Messpunkten rekonstruiert werden.

Das Abtastttheorem (Nyquist-Theorem) besagt, dass dieses Signal mindestens mit der doppelten Frequenz $f_N = 2f_{max}$ abgetastet werden muss, um eine exakte Rekonstruktion zu ermöglichen. In der praktischen Anwendung liefert eine Abtastung mit etwa 3 bis 6 Abtastwerten pro Wellenlänge gute Ergebnisse.

Ist das Abtasttheorem nicht erfüllt (Unterabtastung), gibt es Fehler. Diese werden als Aliasing-Fehler oder Aliasing-Effekte bezeichnet.

4 FFT Algorithmus

4.1 Diskrete Fouriertransformation (DFT)

DFT eines Vektors $\hat{x}$ der Länge N

$$\hat{x}(k) = \sum_{j=0}^{N-1} X(j) \underbrace{W_N^{jk}}_{e^{\frac{2\pi\imath}{N}}}$$

Inverse DFT

$$x(j) = \frac{1}{N} \sum_{j=0}^{N-1} \hat{x}(k)_j W_N^{-jk}$$

Beachte: Inverse kann als DFT der Funktion $\frac{1}{N}\hat{x}(-k)$ berechnet werden.

4.2 Ansatz für schnelle Lösung

Wenn $N = N_1 N_2$ ist, dann kann die DFT 1D-Gleichung als 2D-Gleichung beschrieben werden mit einer Umbenennung der Variablen

$$j = j(a,b) = aN_1 + b, 0 \leq a < N_2, 0 \leq b < N_1$$
$$k = k(c,d) = cN_2 + d, 0 \leq c < N_2, 0 \leq d < N_1$$

Nun wird die Eigenschaft $W_N^{m+n} = W_N^m W_N^n$ ausgenutzt und wir erhalten

$$\hat{x}(c,d) = \sum_{b=0}^{N_1-1} W_N^{b(cN_2+d)} \sum_{a=0}^{N_2-1} X(a,b) W_{N_2}^{ad}$$

Zur Berechnung benötigt man zwei Schritte, zunächst wird die innere Summe berechnet (für alle d)

$$\tilde{x}(b,d) = \sum_{a=0}^{N_2-1} X(a,b) W_{N_2}^{ad}$$

Dazu sind maximal $N_1 N_2^2$ arithmetische Rechenoperationen notwendig. Danach wird die Transformation

$$\sum_{b=0}^{N_1-1} W_N^{b(cN_2+d)} \tilde{x}(b,d)$$

berechnet, was mit $N_1 N_1^2$ Rechenschritten möglich ist.

Der FFT-Algorithmus kann effizient durch in-place Berechnungen kodiert werden, um Rechenschritte und Speicherplatz einzusparen.

4.2.1 Literatur zur FFT

- Cooley, James W., and John W. Tukey, 1965, An algorithm for the machine calculation of complex Fourier series, Math. Comput. 19: 297-301
- Computing in Science and Engineering Jan/Feb 2000, The FFT: An Algorithm the Whole Family Can Use, Daniel N. Rockmore
- E. Oran Brighman, FFT - Schnelle Fourier-Transformation, Einführung in die Nachrichtentechnik, Oldenbourg Verlag, 1982

5 Beispiele

5.1 Amplitudenspektrum

Wir erzeugen ein Test-Signal bestehend aus zwei überlagerten Sinusschwingungen und berechnen das Amplitudenspektrum mittels FFT.

Eine "Frequenz-Verschmierung" ergibt sich dadurch, dass das Signal nicht exakt periodisch ist und Sprünge an den Rändern auftreten.

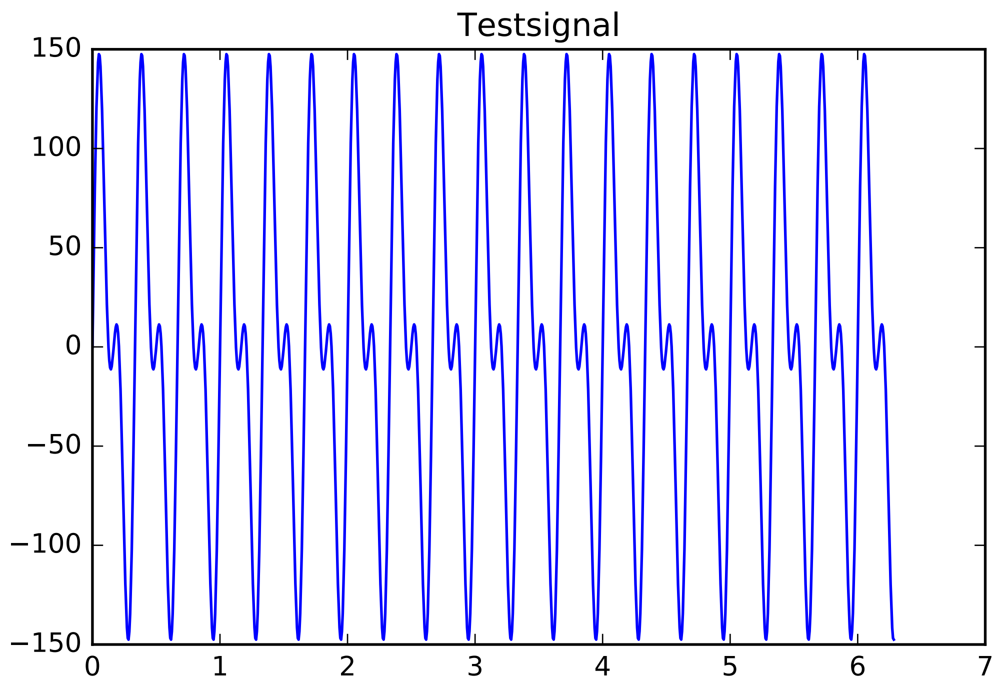
In [49]:
```
%pylab inline
%config InlineBackend.figure_format = 'svg'

M=1000

t = linspace(0, 2*pi, M, endpoint=True)
f1 = 3.0 # Frequenz in Hz
f2 = 6.0 # Frequenz in Hz

A1 = 100.0 # Amplitude
A2 = 70.0 # Amplitude

y = A1 * sin(2*pi*f1*t)+  A2 * sin(2*pi*f2*t)# Signal

plot(t,y)
title('Testsignal')
####################################################
dt = t[1] - t[0] # Abtast-Periode
fa = 1.0/dt # Abtast-Frequenz

Y = fft.fft(y)
```

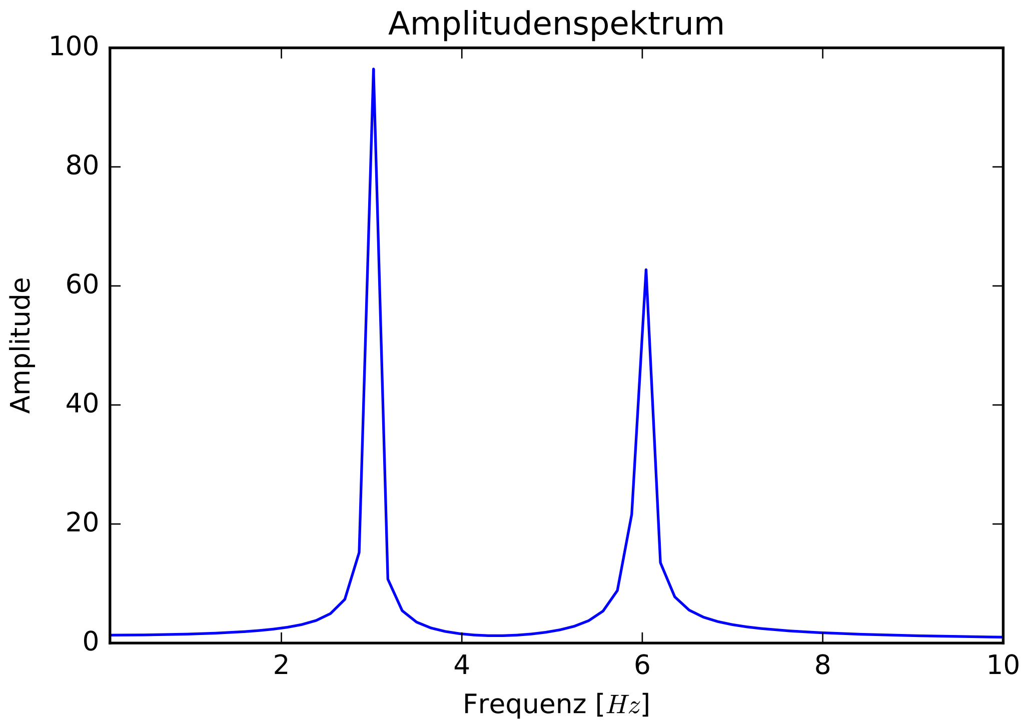
```
N = len(y)/2+1

X = linspace(0, fa/2, N, endpoint=True)

figure()
plot(X,abs(Y[:N])/N)
xlabel('Frequenz [$Hz$]')
ylabel('Amplitude')
xlim([0.1,10])
title('Amplitudenspektrum')
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
/usr/local/lib/python3.4/dist-packages/ipykernel/__main__.py:27: DeprecationWarning: using a non-integer
```

Out[49]: `<matplotlib.text.Text at 0x7fc1a09207b8>`



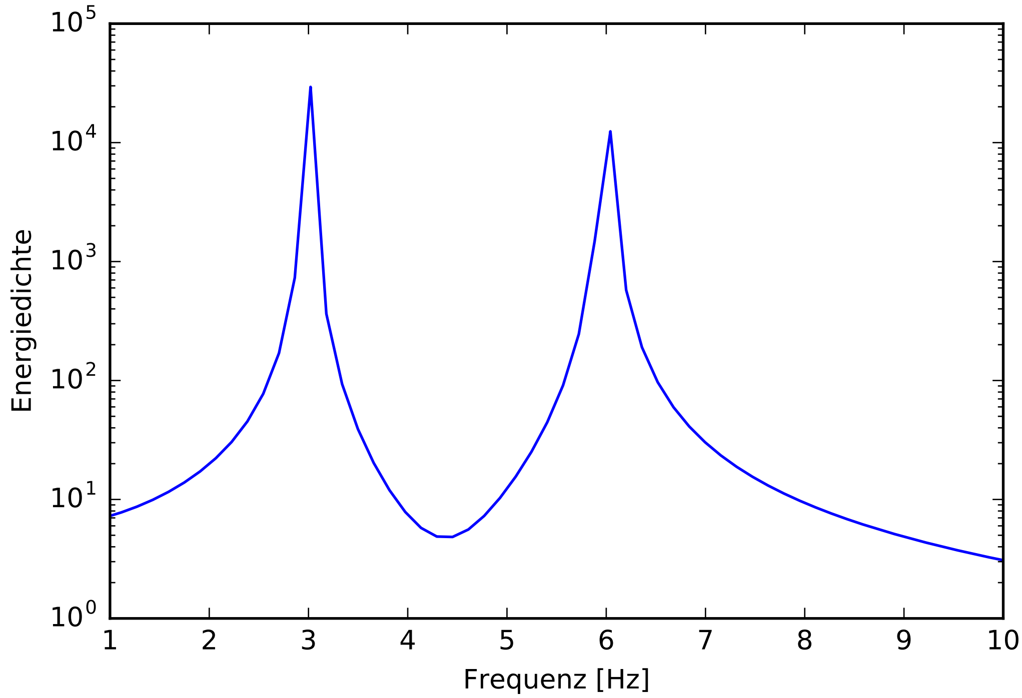


5.2 Periodogramm / Leistunsspektrum

In [50]:
```
f, Pxx = periodogram(y, fa)
figure()
semilogy(f, Pxx)
xlabel('Frequenz [Hz]')
ylabel('Energiedichte')

ylim([1e0, 1e5])
xlim([1, 10])
```

Out[50]:

$$(1, \quad 10)$$

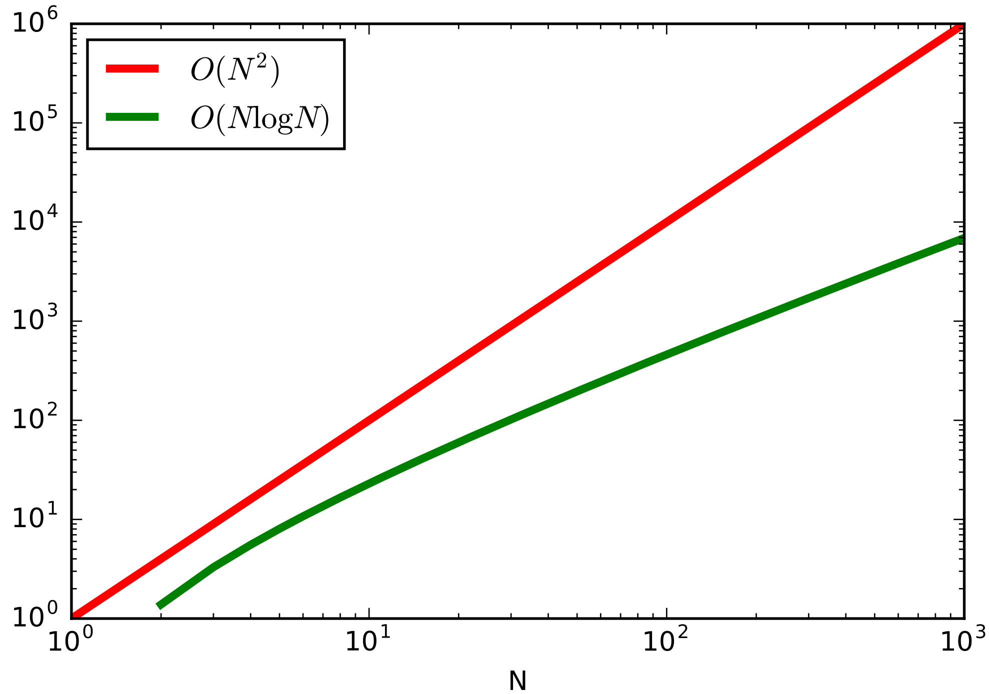


5.3 Komplexität

Der große Vorteil der FFT wird deutlich, wenn man die Anzahl der notwendigen Rechenschritte als Funktion der Vektorlänge N betrachtet.

In [51]:
```
N=1000
n=linspace(1,N,N)
loglog(n,n**2,'r-',linewidth=3,label='$O(N^2)$')
loglog(n,n*log(n),'g-',linewidth=3,label='$O(N \log N)$')
xlabel('N')
legend(loc=2)
```

Out[51]:
```
<matplotlib.legend.Legend at 0x7fc1a078a0f0>
```

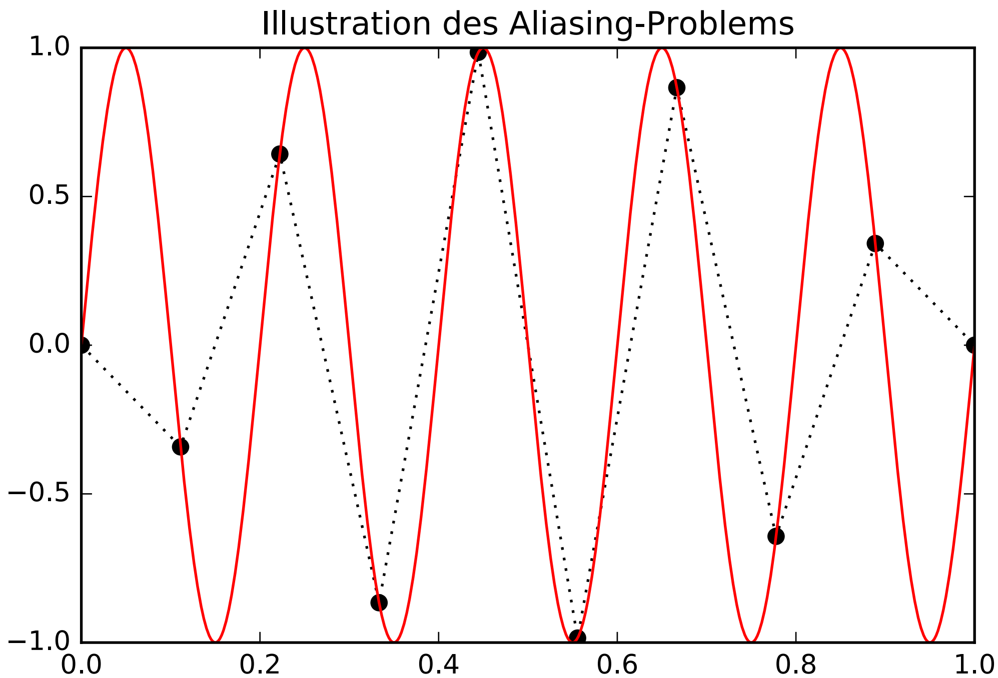


5.4 Alias-Problem

In [52]:
```
T=1.0
f=5.0/T
fN=f*2
print(fN)

N1=10 # Anzahl Abtast-Punkte
N2=1000
##################

t1=linspace(0,T,N1)
y1=sin(2*pi*f*t1)

t2=linspace(0,T,N2)
y2=sin(2*pi*f*t2)

plot(t1,y1,'ko:')
plot(t2,y2,'r-')
title('Illustration des Aliasing-Problems')
```

```
10.0
```

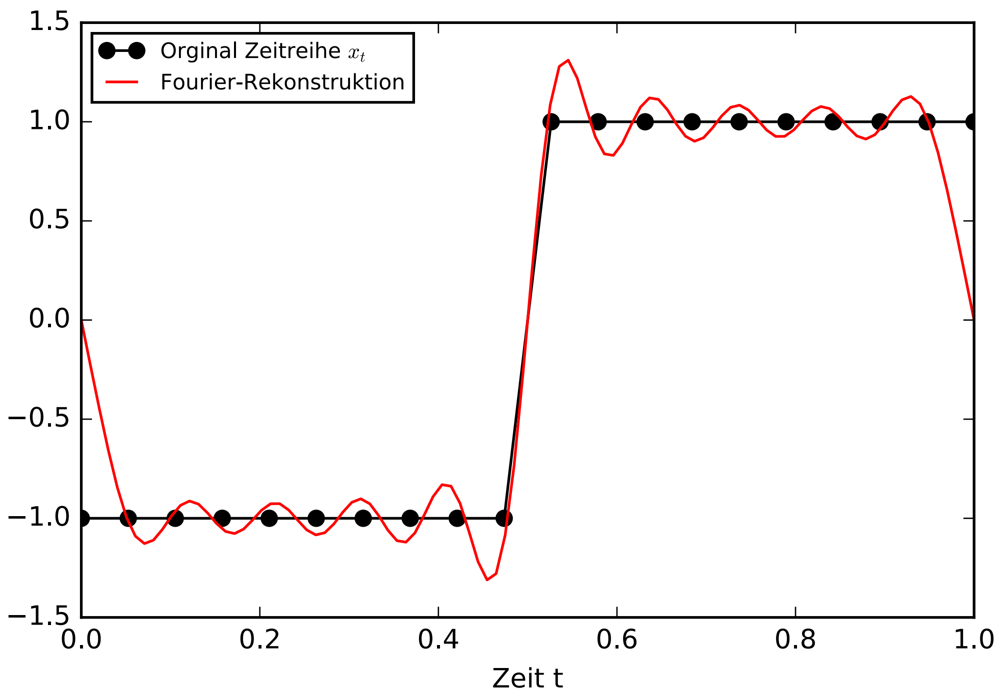
Out[52]:
```
<matplotlib.text.Text at 0x7fc1a08f5a90>
```

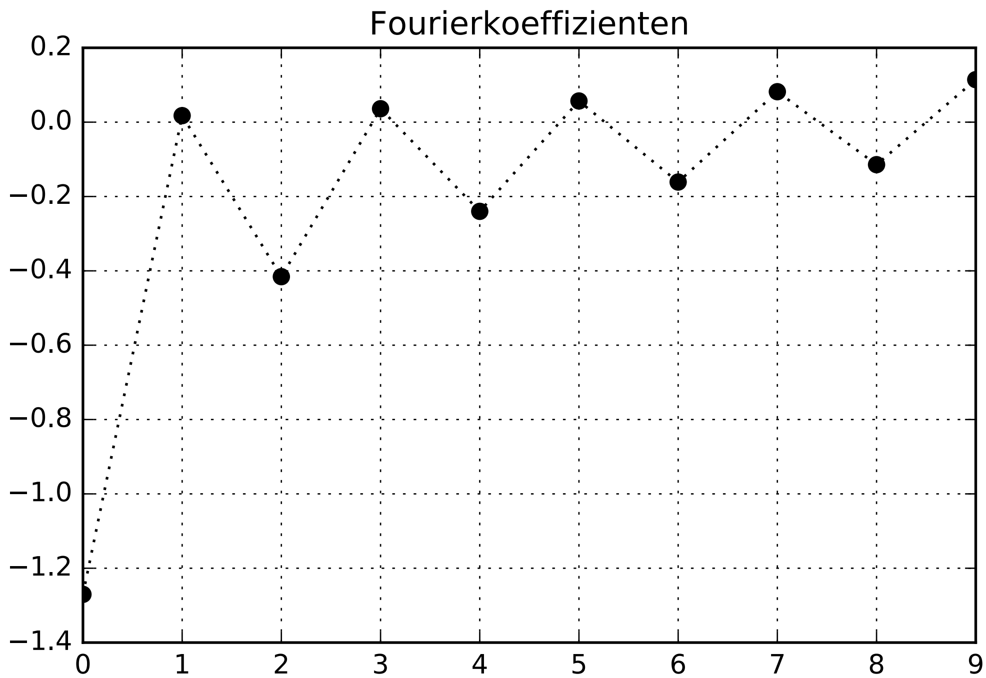


5.5 Berechnung der Fourier-Reihe zu Fuss

In [56]:
```
def x(t):
    case=3 # Beispiele
    if case==1:
        y=sin(t*2*pi) # Reiner Sinus
    if case==2:
        y=sin(t*2*pi) + 0.5*sin(t*8*pi)
    if case==3: # Sprungfunktion
        y=ones(t.size)-2
        y[t>0.5]=1
    return y


N=20
T=1.0

t=linspace(0,T,N)
k=arange(1,N+1)
f_k=k/T

k_max=10 # Nutze Anzahl k_max Fourier-Koeffizienten

ak=zeros(k_max)
bk=zeros(k_max)

# Berechnung der Fourier-Koeffizienten
for i in range(k_max):
```

```
    ak[i]=2/T*sum(x(t)*cos(2*pi*f_k[i]*t))
    bk[i]=2/T*sum(x(t)*sin(2*pi*f_k[i]*t))

figure()
plot(t,x(t),'ko-',label='Orginal Zeitreihe $x_t$')

# Berechnung der Fourier-Reihe (Rekonstruktion)
for i in range(k_max):
    ak[i]=2/T*sum(x(t)*cos(2*pi*f_k[i]*t))/(N-1) # Integral durch Summation approximiert
    bk[i]=2/T*sum(x(t)*sin(2*pi*f_k[i]*t))/(N-1)

t2=linspace(0,T,100) # Neue (größere) Anzahl von Zeitschritten (=Interpolation)

N2=t2.size
x2=zeros(N2)
x2=x2+ak[0]/2
for i in range(k_max):
    x2=x2+ak[i]*cos(2*pi*f_k[i]*t2)+bk[i]*sin(2*pi*f_k[i]*t2)

plot(t2,x2,'r-',label='Fourier-Rekonstruktion')
xlabel('Zeit t')

legend(loc=2,fontsize=8)
figure()

plot(bk,'ko:')
title('Fourierkoeffizienten')
grid()
```




Fourierkoeffizienten
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
−1.4
0
1
2
3
4
5
6
7
8
9