CPII - VL 7 - Teil 2

Signalanalyse

2.1. Entwicklung von Funktionen

2.1 .1 Relle Fourierreihe

$$f_n(x) = \frac{a_0}{2} + \sum_{j=1}^{n} (a_j \cos(k_j x) + b_j \sin(k_j x)), \quad k_j = \frac{j\pi}{b-a}$$

Fourierkoeffizienten : $a_j = \frac{2}{b-a} \int_a^b f(x) \cos(k_j x) \, dx, \quad b_j = \frac{2}{b-a} \int_a^b f(x) \sin(k_j x) \, dx$

$\lim_{n\to\infty} f_n(x) \to f(x)$

Bsp. : a) Dreiecksfunktion : $f_a(x) = 1 - |x|$

b) Sägezahn

c) Rechteck

d) Stufenpuls

2.1 .2 Komplexe Fourierreihe

$$f(x) = \sum_{j=-\infty}^{\infty} c_j e^{i k_j x}, \quad c_j = \frac{1}{b-a} \int_a^b f(x) e^{-i k_j x} \, dx \in \mathbb{C} \; (a_j = c_j + c_{-j}, \; b_j = i(c_j - c_{-j}))$$

2.1 .3 Verallgemeinerungen

A) zeitl. Signale

$$f_n(t) = \frac{a_0}{2} + \sum_{j=1}^{n} (a_j \cos(\omega_j t) + b_j \sin(\omega_j t)) = \sum_{j=-\infty}^{\infty} c_j e^{i \omega_j t}, \quad \omega_j = \frac{j\pi}{T}$$

B) Zerlegung in Fourierkoeffizienten ist eindeutig : Fourier - Transformation

D.h. Ortsraum in Impulsraum, Zeitraum in Frequenzraum (Spektrum)

C) Nicht - periodische Funktionen ($L, T \to \infty$) : kontinuierliche Fouriertransformation

$$c_j = \frac{1}{L} \int_{-L/2}^{L/2} f(x) e^{-i k_j x} \, dx \to \int_{-\infty}^{\infty} f(x) e^{-ikx} \, dx = FT(f(x))(k) = \bar{f}(k)$$

$k_j = j \frac{\pi}{L} \to$ kontinuierliches k

D) Allgemeine Reihenentwicklung : $f(x) =$

$\sum_{n=0}^{\infty} a_n P_n(x)$ z.B. LegendrePolynome : Fourier - Legendre - Reihe

$$a_n = \frac{2n+1}{2} \int_{-1}^{1} f(x) P_n(x) \, dx$$

$$f(x) = \sum_{n=0}^{\infty} \langle n | n \rangle | f \rangle = \sum_{n=0}^{\infty} \langle n | f \rangle | n \rangle = \sum_{n=0}^{\infty} a_n | n \rangle$$

2.2 DiskretFouriertransformation

FT an endlichen N äquidistanten Punkten $\left(a \in C \to \bar{a} \in C\right)$:

$$\bar{a}_k = \sum_{j=0}^{N-1} a_j\, e^{-2\pi i \frac{j k}{N}} \quad (k = 0, \;.., N-1)$$

Anwendungen :

Frequenzanalyse / Spektrumanalyse

Filter, etc.

Rücktranformation : $a_j = \frac{1}{N} \sum_{k=0}^{N-1} \bar{a}_k\, e^{2\pi i \frac{j k}{N}}$

Berechnung so sehr aufwendig $\left(N^2\right)$: FFT (Fast Fourier Transformation Algorithmen)

Cooley - Turkey, 1965 : Divide - Konquer - Algorithmus : O (N log (N))

DFT verstehen

Samplefrequenz : $\nu = \frac{1}{\Delta T}$, maximale Frequenz (Nyquist - Frequenz) : $\nu_{Ny} = \frac{\nu}{2} = \frac{1}{2\,\Delta T}$

Bsp : T = 2 s, 200 Samples $\to$ ΔT = 10 ms : ν = 100 Hz, ν_{Ny} = 50 Hz

Python : fftshift (fft (Signal))
oder
ν = fftfreq (N, ΔT)

Beispiele :

1) Reine Schwingung
2) Gedämpfte Schwingung
3) Gedämpfte Schwingung → Einfluß der Samplepunkte