

2.3 .3 Signalanalyse

- Detrend
- PeakFind (Maxima finden)
scipy.signal.find_peak_cwt ()
- Periodogramm (gleitende Fouriertransformation) -> Zeitreihenanalyse
PSD - power spectral density | FT (Signal) | ^2
scipy.signal.periodogram ()
- Welch - Methode : Unterdrückung des Rauschens (Spezielle Fensterfunktion und Mittelung)
scipy.signal.welch ()

2.3 .4 Fourier - Filter

Tiefpassfilter, Hochpassfilter
Anti-Aliasdurch vorgeschalteten TP Filter

2.3 .5 Bildbearbeitung

- Streifen (Noise) Filtern -> TP Filter
- Weichzeichnen -> 1. Scharfe Filter (z.B. TP Filter) : "Ringing"
2. Glatte Filter : Gauss - Filter, Butterworth - Filter
- Kantenfilter -> HP Filter
- Schärfen von Bildern : Unschärferes Bild (TP Filter) vom Originalbild abziehen
-> Kanten werden betont, bessere Wahrnehmung

2.4 Faltung

$$(f * g) (t) = \int_{-\infty}^{\infty} f (\tau) g (t - \tau) d\tau$$

$$(f * g) _n = \sum_{k=1}^{\infty} f_k g_{n-k}$$

Berechnung : Faltungstheorem : FT (f * g) = FT (f) FT (g)

d.h. f * g = IFT (FT (f) FT (g))

Anwendung : Glättung

f (t) glättet mit g (t) ("Gauss-Funktion") -> hohen Frequenzen gefiltert

numpy .convolve ()

scipy.signal.convolve ()

scipy.signal.fftconvolve ()

Anwendung : Linienverbreiterung von Spektrallinien

* Dopplerverbreiterung (Gaussverteilung)

* Druckverbreiterung (Lebensdauer von Zuständen, Lorentzverteilung)

Gauss * Lorentz = Voigt-Profil