

0. Grundlagen

Die Fourier-Analyse ist die **Zerlegung eines Zeitsignals in Sinusoiden zunehmender Frequenzen**, sodass durch deren Summierung (=Fourier-Synthese) das Zeitsignal rekonstruiert wäre. Durch die Fourier-Analyse bekommt man ein **Spektrum**, das die Amplituden-Werte der Sinusoiden (in dB) als Funktion der Frequenz enthält.

Die digitale Anwendung der Fourier-Analyse wird an hand einer **FFT** (Fast-Fourier-Transform) durchgeführt.

fs die Abtastrate vom Signal in Hz

N die Länge von dem digitalen Zeitsignal (auf das der FFT angewandt wird) in Punkten.

0.1 Konvertierung zwischen Punkten und ms

(1) Ein Signal von **N** Punkten hat eine Dauer von **1000 N/fs** ms

z.B. **N** = 256 Punkte, **fs** = 16000 Hz. Dauer, **d**, = $1000 \times 256 / 16000 = 16$ ms

(2) Ein Signal von **d** ms enthält **(d fs)/1000** Punkte.

z.B. **d** = 32 ms, **fs** = 10000 Hz, **N** = $32 \times 10000 / 1000 = 320$ Punkte.

0.2 Eigenschaften eines Spektrums

Nach der Anwendung einer FFT hat ein Spektrum folgende Eigenschaften

(3) Höchst sichtbare Frequenz, **fmax** = **fs/2** Hz.

(4a) Frequenzauflösung **fres** = Abstand zwischen Spektralkomponenten = **fs/N** Hz
daher

(4b) **N** = **fs/fres** Punkte

(5) Anzahl der Spektralkomponente **fnum** = **(N/2) + 1**

0.3 Anwendung einer FFT

Um ein FFT anzuwenden, muss **N** festgelegt werden (= die Anzahl der Punkte im digitalen Zeitsignal, auf das die FFT angewandt wird) und **N** von einer Potenz 2 sein (z.B. 64, 128, 256, 512... Punkte).

Die anderen Eigenschaften vom Spektrum werden dann wie oben durch **fs** und **N** festgelegt.
zB

fs = 10000, **N** = 512 (daher ist übrigens die Fensterdauer $1000 \times 512 / 10000 = 51.2$ ms)

fmax = 5000 Hz

fres = $10000 / 512 = 19.53125$ Hz

fnum = $(N/2) + 1 = 257$.

Also 257 dB Werte zu Frequenzen 0 Hz, 19.53125 Hz, 39.06250 Hz, 58.5937 Hz ...
4980.46875 Hz 5000 Hz

Oft wird **N** indirekt über **fres** festgelegt: in dem Fall wird meistens der nächst liegende **N** einer Potenz 2 genommen (und **fres** dann dementsprechend umgesetzt)

z.B. Anwendung von einer Fourier-Analyse **fs** = 16000 Hz, **fres** = 40 Hz.

N = $16000 / 40 = 400$ Punkte. (4a)

Der nächst liegende **N** einer Potenz 2 ist 512 Punkte.

Daher

$$\mathbf{fmax} = 8000 \text{ Hz} \quad (3)$$

$$\mathbf{fres}: 16000/512 = 31.25 \text{ Hz} \quad (4)$$

$$\mathbf{fnum}: 512/2 + 1 = 257 \quad (5)$$

0.4 Zusammenfassung

Gegeben $\mathbf{fs}$ und $\mathbf{N}$

$$\mathbf{d}: = 1000 \text{ N/fs} \quad \text{ms}$$

$$\mathbf{fmax} = \mathbf{fs}/2$$

$$\mathbf{fres} = \mathbf{fs}/\mathbf{N}$$

$$\mathbf{fnum} = (\mathbf{N}/2) + 1$$

1. Anwendung in Emu-R

tkassp, Spektrum, Perform-Analysis.

1.1 Parameter

fres = Frequency resolution = Frequenzauflösung (siehe oben)

Window shift = wie oft (mit welchem zeitlichen Abstand) sollen Spektra berechnet werden?

1.2 Default Parameter, die meistens nicht geändert werden müssen

- FFT Length = $\mathbf{N}$ (wird anhand **fres** vermittelt; siehe 0.3)
- LP-Type = DFT um Spektra zu berechnen
- Window-Function = Blackman, um das Sprachsignal vor der DFT-Berechnung mit einem Cosinus-Fenster zu glätten (siehe Harrington, 2010 Kap. 8 für Details)
- Output-extension: dft (diese Extension muss dann in Emu unter Tracks eingetragen werden)

1.3 Emu-R Grundbefehle um Spektra zu manipulieren

[ç, x] Frikative der Sprachdatenbank `timetable`, Ebene = `Phonetic` ("C", "x")

`dor = emu.query("timetable", "*", "Phonetic = C | x")`

Label-Vektor

`dor.l = label(dor)`

Spektra fuer die Segmentliste

`dor.dft = emu.track(dor, "dft")`

Spektra zum zeitlichen Mittelpunkt

`dor.dft5 = dcut(dor.dft, .5, prop=T)`

Wie erkennt man $\mathbf{N}$, $\mathbf{d}$, $\mathbf{fmax}$, $\mathbf{fres}$, $\mathbf{fnum}$? (siehe 0.4)

fnum: Anzahl der Spektralkomponente

`ncol(dor.dft)`

`ncol(dor.dft5)`

$\mathbf{N}$: (FFT length)

```

N = 2 * (ncol(dor.dft)-1)

# Die Frequenzen, zu denen die Spektralkomponente vorkommen
trackfreq(dor.dft)

# fmax
max(trackfreq(dor.dft))

# Daher die Abtastrate, fs
fs = 2 * max(trackfreq(dor.dft))

# fres: Hz-Abstand zwischen den Spektralkomponenten
fs/N

# Auch durch
diff(trackfreq(dor.dft))

# d (Dauer des Fensters, mit der die Spektra berechnet wurden)
1000*N/fs

```

1.4 Abbildungen

```

# Prüfen ob es sich um ein Spektrales-Object handelt
class(dor.dft5)

# Alle Spektra (zum zeitlichen Mittelpunkt)
plot(dor.dft5)

# Alle Spektra nach Etikettierung kodiert
plot(dor.dft5, dor.l)

# Ensemble-averaged Spektra
plot(dor.dft5, dor.l, fun=mean, dbnorm=T)

# Spektra zum zeitlichen Mittelpunkt zwischen 1000-2000 Hz
dor.dft5[,1000:2000]

# das gleiche, Segmente 4, 5, 7 Frequenzen 2000-2500 Hz
dor.dft5[c(4,5,7), 2000:2500]

# Abbildung, Spektra, pro Kategorie gemittelt, 500 - 5000 Hz
plot(dor.dft5[,500:5000], dor.l, fun=mean, dbnorm=T)

```