Einführung in die Signalverarbeitung – Übung I

Phonetik und Sprachverarbeitung, 2. Fachsemester, Block Sprachtechnologie I

Florian Schiel

Institut für Phonetik und Sprachverarbeitung, LMU München

Signalverarbeitung - Teil 8

Allgemeines

- Unterrichtssprache ist Deutsch (englische Fachbegriffe in Klammern)
- Fragen am besten sofort; besser einmal zuviel gefragt
- Literatur:
  - Jurafsky D, Martin J H (2000): Speech and Language Processing. Prentice Hall, Kap I.7.
  - Schrüfer E (1980): Signalverarbeitung
  - Pfister B, Kaufmann T (2008): Sprachverarbeitung - Grundlagen und Methoden der Sprachsynthese und Spracherkennung. Springer-Verlag Berlin Heidelberg.
  - Rabiner, Lawrence R., Schafer R W (1978): Digital Processing of Speech Signals. Prentice-Hall, New Jersey, USA.
  - Hess W (1993): Digitale Filter. Teubner Studienbücher, B.G.Teubner, Stuttgart.
  - Harrington J, Cassidi St (1999): Techniques in Speech Acoustics. Kluwer Academic Publishers, Dordrecht/Boston/London.

Übungen mit R I

- Signalausschnitt lesen
- Fensterung
- DFT
- Cepstrum berechnen
- Grundfrequenz aus Cepstrum bestimmen
- Cepstrum liftern

In Zweiergruppen an einem Linux-Rechner einloggen:

- Einloggen mit account ’ekurs2’ oder eigenem Account
- Zwei Terminal-Fenster starten (auswählen: `konsole`)

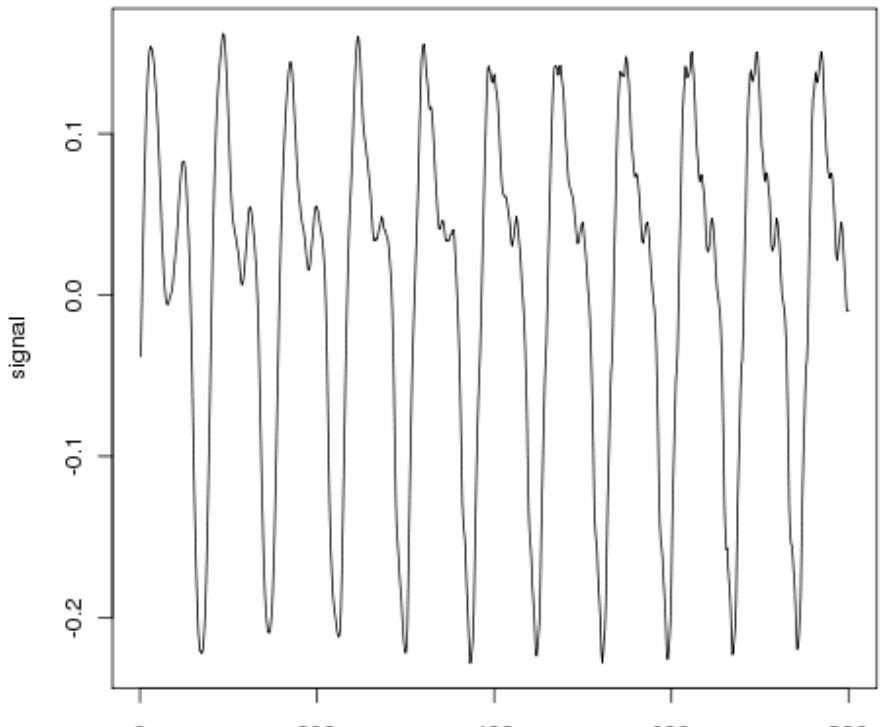
Signalausschnitt bestimmen und einlesen I

(based on praat version 5.3.16)

- Starten Sie praat (Terminal-Fenster starten (`konsole`), eingeben: `praat`)
- Laden Sie das File `/bmnt/BAS/PD2/data/awe/awed5240.nis` (auswählen: `Read / Read from file...`)
- Finden Sie den Zeitpunkt eines Vokals (auswählen: `Edit`), z.B. 1.61sec
- Auswählen: `Convert / Extract part ...`, geben Sie einen `Time range` von 50msec um dem gewählten Zeitpunkt ein (z.B. 1.58 - 1.63) und drücken `Ok`
- Speichern Sie das ausgewählte Sprachsignalstück in eine Textdatei (auswählen: `Save / Save as short text file ...`), z.B. in `awed.Sound`
- Löschen Sie die ersten 13 Zeilen des Textfiles:
  `tail -n +14 awed.Sound > awed.txt`

Signalausschnitt bestimmen und einlesen II

- R starten: `R`
- Signalstück als Vektor einlesen:
  ```
  signal = read.table("awed.txt")
  signal = signal$V1
  ```
- Signalstück plotten:
  ```
  plot(signal,type="l")
  ```



Signalstück mit Hamming fenstern I

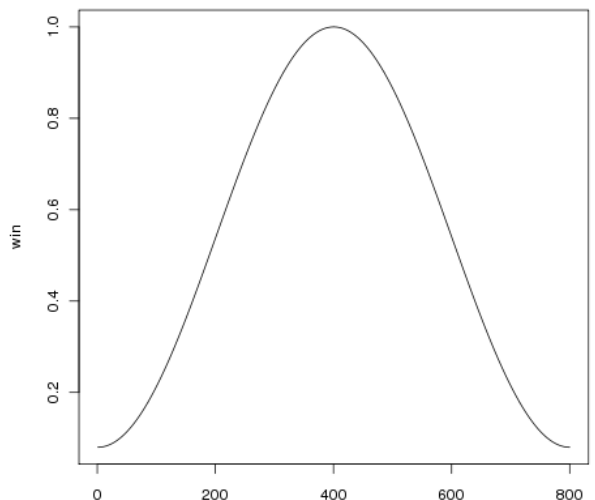
Zunächst das Fenster nach Hamming erzeugen:

- Gleich langes Signal von $-\pi$ bis $+\pi$:

```
rad = seq(from=-pi,to=pi,length.out=length(signal))
rad
-3.141593 -3.133729 -3.125865 -3.118001 -3.110137 -3.102274
-3.094410 -3.086546 -3.078682 -3.070818 -3.062955 -3.055091
-3.047227 -3.039363 -3.031499 -3.023635 -3.015772 -3.007908
-3.000044 -2.992180 ...
```

- Window-Signal berechnen:

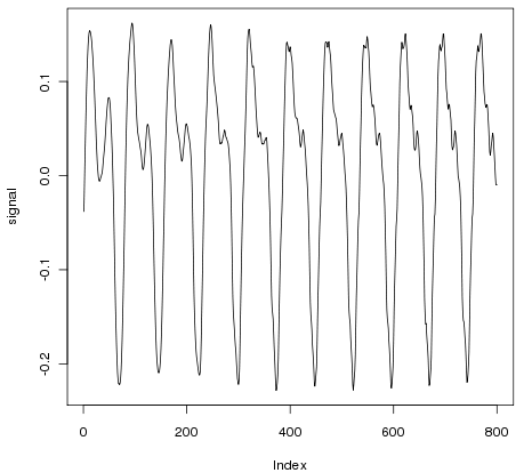
```
win = 0.54 + 0.46 * cos(rad)
plot(win,type="l")
```

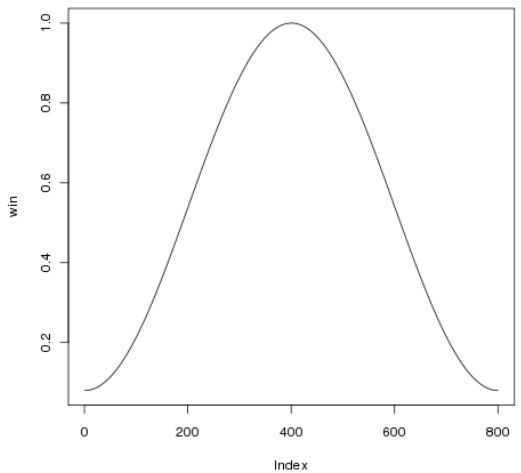


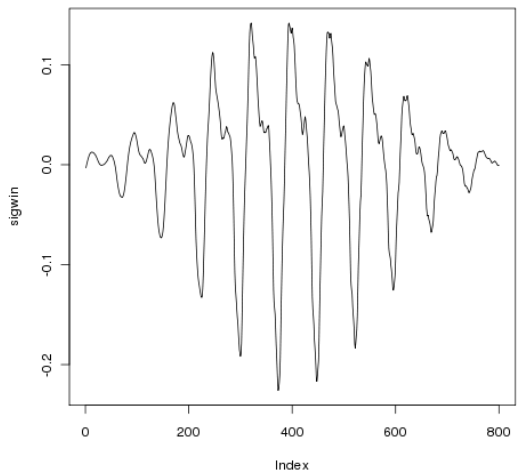
Signalstück mit Hamming fenstern II

- Signalstück mit Fenster multiplizieren:
  `sigwin = signal * win`
- Alle drei Signale übereinander darstellen:

```
par(mfrow=c(3,1))
plot(signal,type="l")
plot(win,type="l")
plot(sigwin,type="l")
par(mfrow=c(1,1))
```







Diskrete Fouriertransformation I

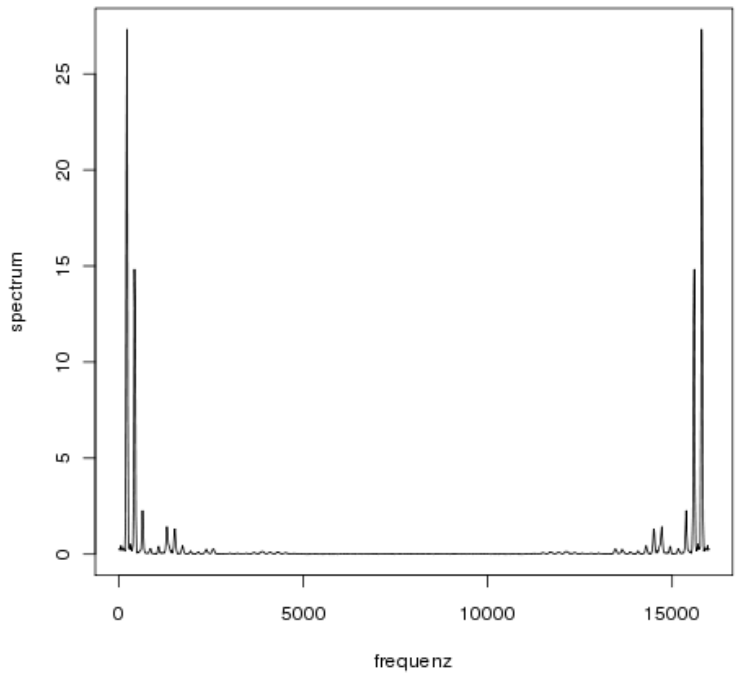
Fouriertransformierte des gefensterten Signalstücks:

- ```
  dft = fft(sigwin)
  length(dft)
  800
  class(dft)
  "complex"
  dft[1:10]
  0.2360764+0.0000000i 0.2769177-0.0083768i
  0.2122624+0.0468455i 0.4384431+0.0409965i
  0.1411284-0.1610980i 0.2747270+0.1272546i
  0.2614101+0.0135853i 0.3159110+0.0212894i
  0.1197222+0.1065230i -0.8949012+1.5977952i
  ```
  Beachte: Komplexe Zahlen Re + Im * i
- Betragsspektrum berechnen: $|S| = \sqrt{Re^2 + Im^2}$
  ```
  spectrum = abs(dft)
  spectrum[1:10]
  0.2360764 0.2770443 0.2173702 0.4403556 0.2141724 0.3027683
  0.2617629 0.3166276 0.1602515 1.8313377
  ```
  ```

Diskrete Fouriertransformation II

- Betragsspektrum enthält erst posititve, dann negative Frequenzen:
  `plot(spectrum,type="l")`
- Frequenzen der Abtastwerte im Spektrum berechnen:

```
len = length(spectrum)
frequenz = seq(from=0,to=16000,length.out=len)
plot(frequenz,spectrum,type="l")
```



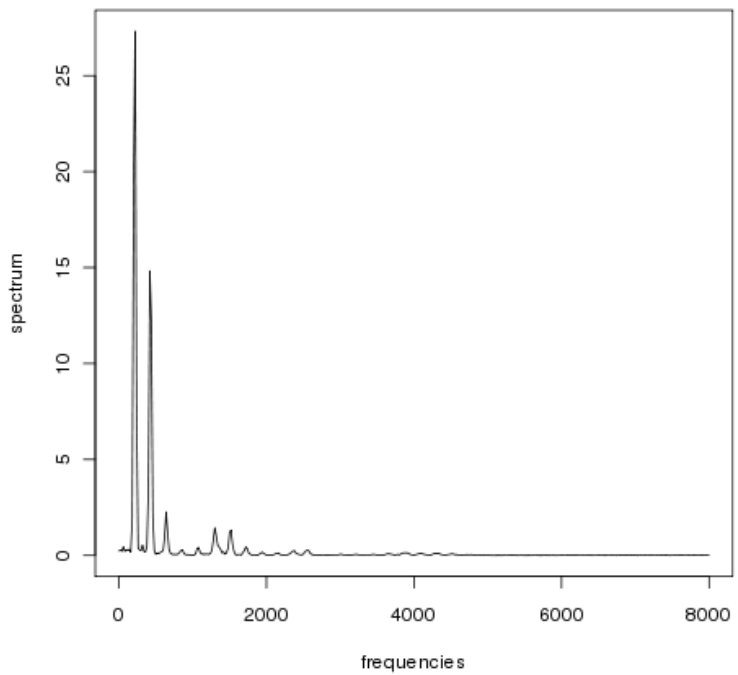
Diskrete Fouriertransformation III

- Abtastrate = 16000Hz -> positives Spektrum von 0 bis 8000Hz:
- Spektrum auf die positiven Frequenzen beschränken:

```
spectrum = spectrum[1:(len/2)]
frequency = frequenz[1:(len/2)]
```

- Spektrum mit Frequenzindex plotten:

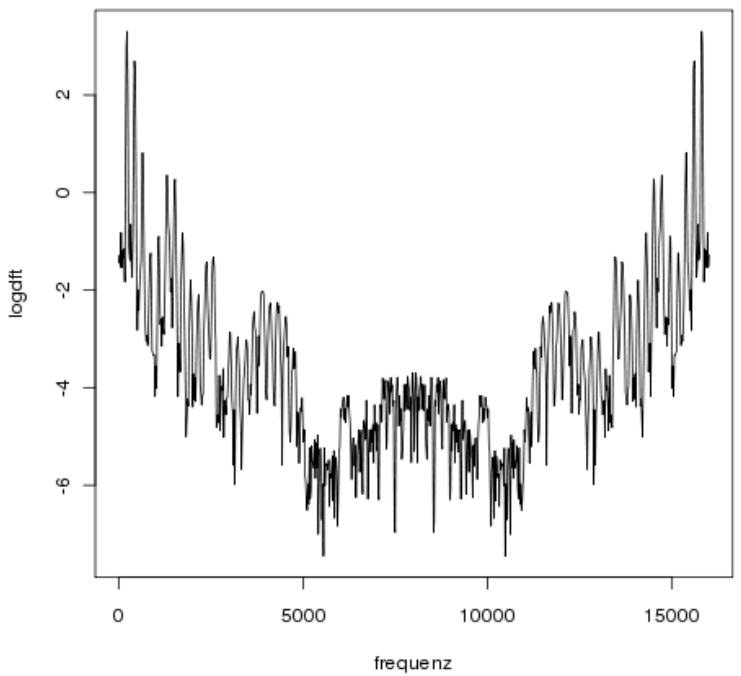
```
plot(frequency,spectrum,type="l")
```



Cepstrum berechnen I

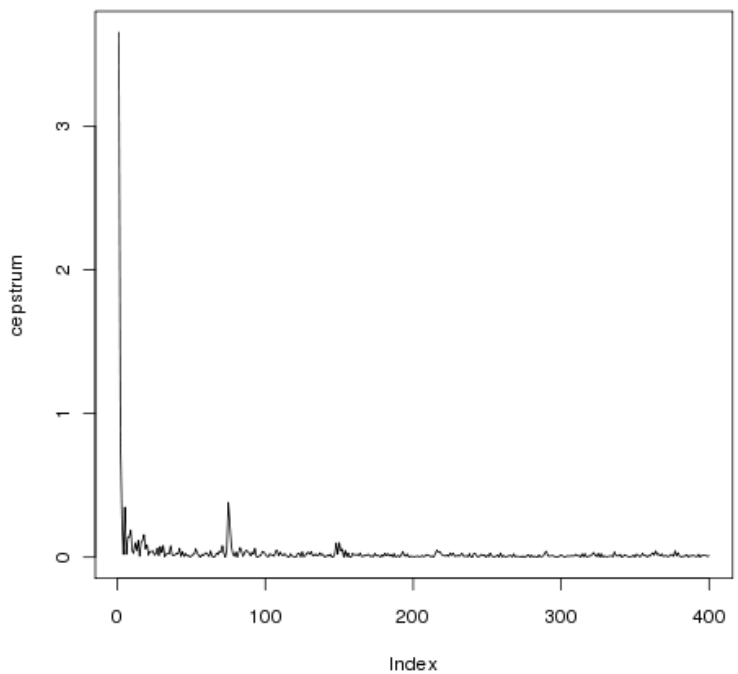
- Betragsspektrum logarithmieren:

```
logdft = log(abs(dft))
plot(frequenz,logdft,type="l")
```



Cepstrum berechnen II

- Rücktransformation:
  `cep = fft(logdft,inverse=T) / length(logdft)`
- Cepstrum ist symmetrisch wie Spektrum: nur die erste Hälfte:
  ```
  cepstrum=abs(cep[1:(length(cep)/2)])
  plot(cepstrum,type="l")
  ```



Grundfrequenz aus Cepstrum bestimmen I

Das erste ausgeprägte Maximum im Cepstrum repräsentiert die harmonische Struktur des Signals und damit die Grundfrequenz f_0

- Position des Maximums finden: im Bereich 60-90 samples:

```
cepstrum[50:100]
-15.8772765 4.9430562 -34.9959917 6.0302169 -0.3280844
-4.7643822 -21.9910494 14.9209326 -34.2420543 -27.1405632
63.3247850 22.9449767 -0.9348028 44.8737785 305.2771011
188.9483104 64.5263052 -19.0631694 6.2700516 -27.7649039
-0.9054017 26.8393667 52.8060962 31.3898429 5.4932176
24.8430083 -38.0053984 -32.9210344 -21.9340710 -7.9960797
-27.9268497
```

  -> sample 50 + 15 = samlple 65 hat Maximum

- Abtastrate 16000Hz -> $65 samples = 65/16000 sec$

```
65/16000
0.0040625
```

- Grundperiodendauer $T_0 = 0.0040625 sec$ entspricht $f_0 = \frac{1}{T_0} Hz$

```
1/0.0040625
246.1538
```

  -> $f_0 = 246 Hz$
  Plausibel, weil es sich um eine Frauenstimme handelt.

Cepstrum 'liftern' I

Für die Berechnung der Einhüllenden des Spektrums (Formanten) kann man versuchen, das Cepstrum auf die unteren Werte, welche die Grobstruktur des Spektrums repräsentieren, zu beschränken und wieder in den Spektralbereich zurücktransformieren.
Diese Technik nennt man 'liftern' (von 'filtern').

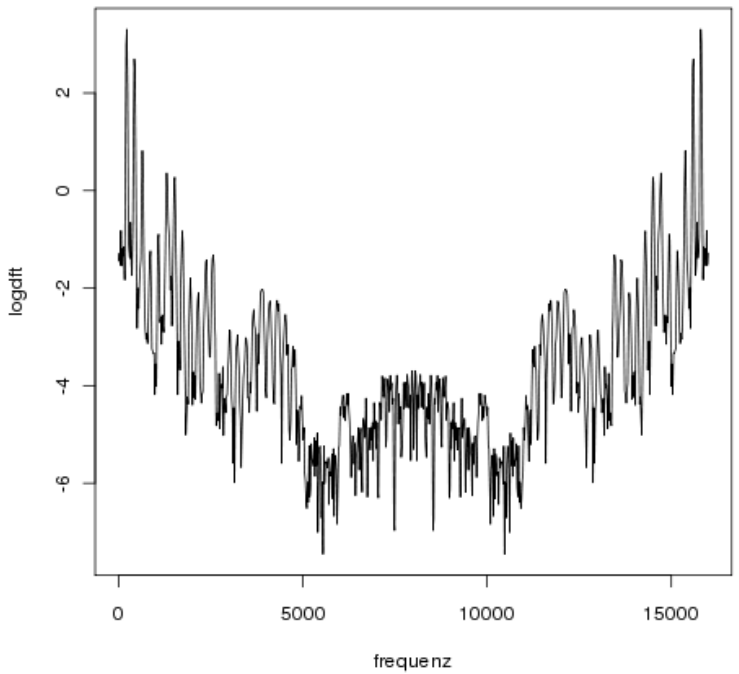
- Im (symmetrischen) Cepstrum ab dem Abtastwert 20 alles zu 0 setzen:
  `cep[20:782] = 0`
- Kontrolle:

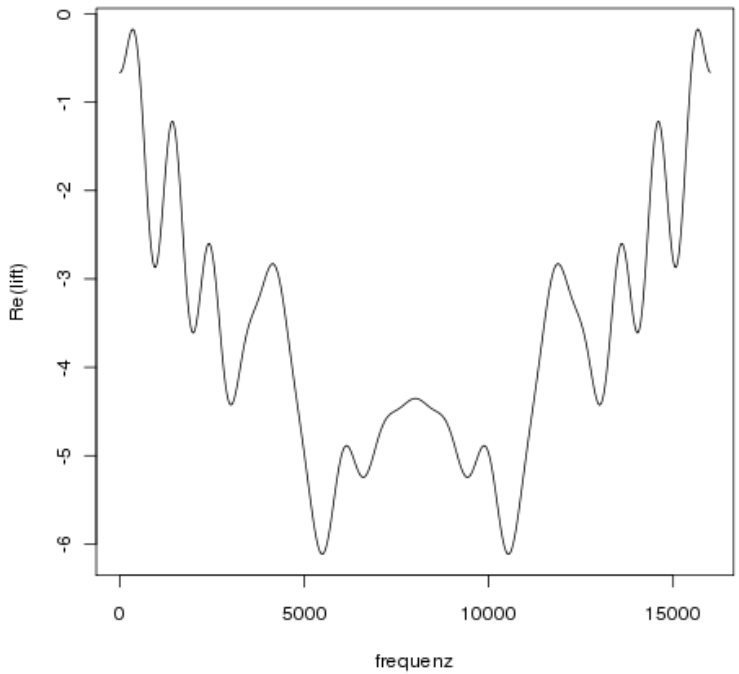
```
cep[10:21]
-0.043513324-0i 0.028343669+0i 0.099614371+0i
0.049229900-0i 0.115774387+0i -0.007235267+0i
-0.102762772+0i -0.116052162-0i -0.156117844+0i
-0.058182236-0i 0.000000000+0i 0.000000000+0i
cep[780:790]
0.000000000+0i 0.000000000+0i 0.000000000+0i
-0.058182236+0i -0.156117844-0i -0.116052162+0i
-0.102762772-0i -0.007235267-0i 0.115774387-0i
0.049229900+0i 0.099614371-0i 0.028343669-0i
```

Cepstrum 'liftern' II

- Cepstrum wieder transformieren:

```
lift = Re(fft(cep))
par(mfrow=c(2,1))
plot(frequenz,logdft,type="l") plot(frequenz,lift,type="l")
```





Cepstrum 'liftern' III

- Cepstral geglättetes Spektrum:

```
plot(frequency,lift[1:(len/2)],type="l")
plot(frequency,logdft[1:(len/2)],type="l")
```