Einführung in die Signalverarbeitung – Übung II

Phonetik und Sprachverarbeitung, 2. Fachsemester, Block Sprachtechnologie I

Florian Schiel

Institut für Phonetik und Sprachverarbeitung, LMU München

Signalverarbeitung - Teil 9

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 II

- Zero-Padding im Zeitbereich
- Zero-Padding im Frequenzbereich
- Höhenanhebung (Pre-Emphase)

An einem Linux-Rechner einloggen:

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

Zero-Padding im Zeitbereich I

- Signalstück laden:

```
signal = read.table("awed.txt")
signal = signal$V1
```

- Signal mit Hamming fenstern:

```
rad = seq(from=-pi,to=pi,length.out=length(signal))
win = 0.54 + 0.46 * cos(rad)
sigwin = signal * win
```

- Amplitudenspektrum berechnen:

```
fabt = 16000
len = length(sigwin)
dft = fft(sigwin)
spectrum = abs(dft[1:(len/2)])
frequenz = seq(from=0,to=fabt/2,length.out=len/2)
plot(frequenz,spectrum,type="l")
```

Zero-Padding im Zeitbereich II

- Signalstück (gefenstert) mit Nullen auf 4-fache Länge verlängern:

```
sigzero = c(sigwin,rep(0,3*len))
plot(sigzero,type="l")
```

- Amplitudenspektrum vom zero-pad berechnen:

```
len = length(sigzero)
dft = fft(sigzero)
speczero = abs(dft[1:(len/2)])
freqzero = seq(from=0,to=fabt/2,length.out=len/2)
par(mfrow=c(2,1))
plot(frequenz,log(spectrum),type="l")
plot(freqzero,log(speczero),type="l")
```

- Auf ersten Blick kein Unterschied, aber bei kleinerem Frequenzbereich:

```
plot(frequenz,log(spectrum),xlim=c(0,1000),type="l")
plot(freqzero,log(speczero),xlim=c(0,1000),type="l")
```

Zero-Padding im Frequenzbereich I

- Neues Signalstück (50msec) mit Frikativ extrahieren.
- Signalstück laden:

```
signal = read.table("Frikativ.txt")
signal = signal$V1
```

- Signal mit Hamming fenstern:

```
rad = seq(from=-pi,to=pi,length.out=length(signal))
win = 0.54 + 0.46 * cos(rad)
sigwin = signal * win
plot(sigwin,type="l")
plot(sigwin,type="l",xlim=c(380,420))
```

  Signal ist deutlich 'eckig'.
- Amplitudenspektrum berechnen:

```
fabt = 16000
len = length(sigwin)
dft = fft(sigwin)
spectrum = abs(dft[1:(len/2)])
frequenz = seq(from=0,to=fabt/2,length.out=len/2)
plot(frequenz,spectrum,type="l")
```

Zero-Padding im Frequenzbereich II

- Komplexes Spektrum mit Nullen auf vierfache Länge verlängern:
  VORSICHT: Die symmetrische Struktur des komplexen Spektrums muss erhalten bleiben!
  D.h. die Nullen müssen in der Mitte hinzugefügt werden.
  ```
  speczero = c(dft[1:(len/2)],rep(0+0i,3*len),dft[((len/2)+1):len])
  plot(log(abs(dft)),type="l")
  plot(log(abs(speczero)),type="l")
  ```
- Zero-padded Spektrum zurücktransformieren:
  ```
  sigzero = Re(fft(speczero,inverse=T) / len)
  plot(sigzero,type="l")
  plot(sigzero,type="l",xlim=c(4*380,4*420))
  ```
  Signal ist deutlich höher abgetastet (interpoliert).
  Zum Vergleich nochmal das ursprüngliche Signal:
  ```
  plot(sigwin,type="l",xlim=c(380,420))
  ```

Höhenanhebung (pre-emphasis) I

- Signalstück laden:

```
signal = read.table("awed.txt")
signal = signal$V1
```

- Signal mit Hamming fenstern:

```
rad = seq(from=-pi,to=pi,length.out=length(signal))
win = 0.54 + 0.46 * cos(rad)
sigwin = signal * win
```

- Amplitudenspektrum berechnen:

```
fabt = 16000
len = length(sigwin)
dft = fft(sigwin)
spectrum = abs(dft[1:(len/2)])
frequenz = seq(from=0,to=fabt/2,length.out=len/2)
plot(frequenz,spectrum,type="l")
```

Höhenanhebung (pre-emphasis) II

- Pre-Emphase durchführen:

```
sigpre = signal
sigpre[2:len] = signal[2:len] - 0.98 * signal[1:(len-1)]
sigprewin = sigpre * win
```

- Spektrum berechnen von Pre-Emphase:

```
dftpre = fft(sigprewin)
spectrumpre = abs(dftpre[1:(len/2)])
par(mfrow=c(2,1))
plot(frequenz,spectrum,type="l",main="Ohne Höhenanhebung")
plot(frequenz,spectrumpre,type="l",main="Mit
Höhenanhebung")
par(mfrow=c(1,1))
```